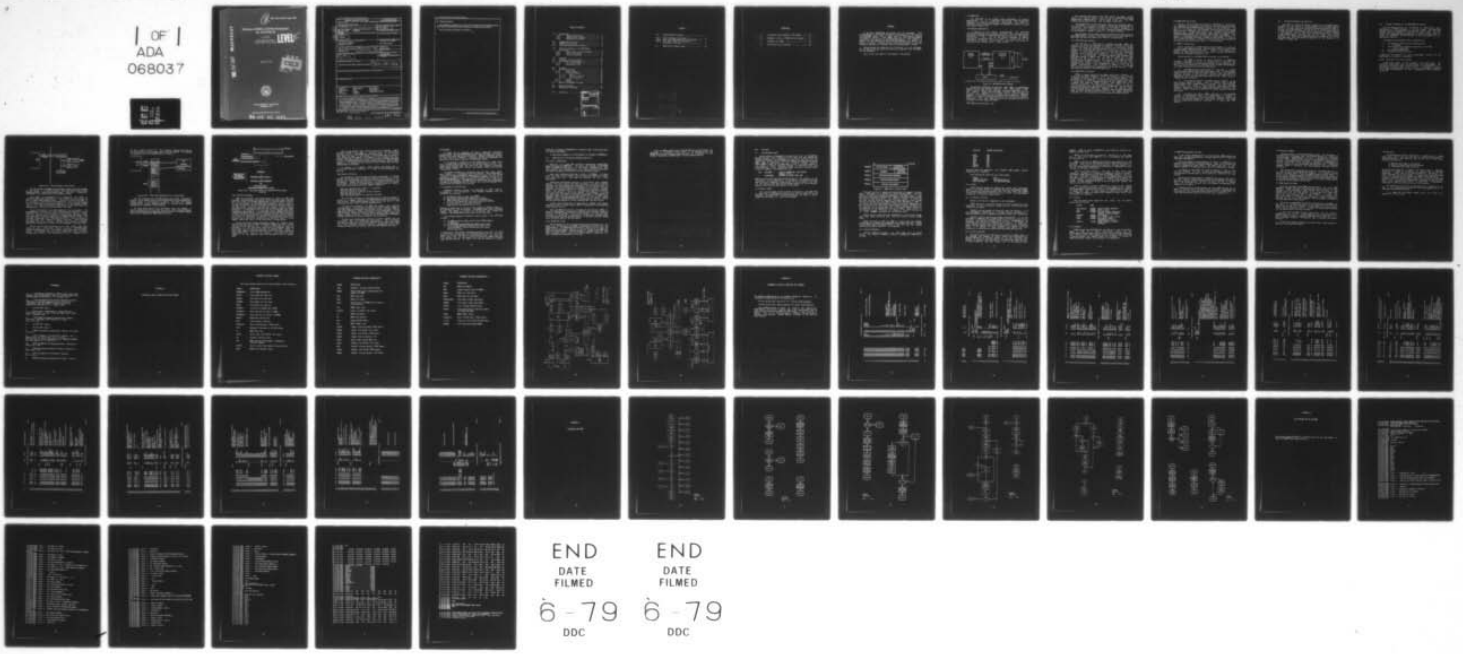
AD-A068 037 NAVAL RESEARCH LAB WASHINGTON D C F/G 17/2
MULTIPORT RS232 TO NTDS MULTIPLEXOR FOR AN/UYK-20.(U)
APR 79 L E RUSSO
UNCLASSIFIED NRL-MR-3980 NL

| OF |
ADA
068037



END
DATE
FILMED
6-79
DDC

END
DATE
FILMED
6-79
DDC

NRL Memorandum Report 3980

Multiport RS232 to NTDS Multiplexor for AN/UYK-20

L. E. RUSSO

Signal Exploitation Branch
Communications Sciences Division

LEVEL II

April 23, 1979

NAVAL RESEARCH LABORATORY
Washington, D.C.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: NRL Memorandum Report 3980
2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle): MULTIPORT RS232 TO NTDS MULTIPLEXOR FOR AN/UYK-20
5. TYPE OF REPORT & PERIOD COVERED: Final report
6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s): L. E. Russo
8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS: Naval Research Laboratory, Washington, DC 20375
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: NRL Problem B02-20

11. CONTROLLING OFFICE NAME AND ADDRESS: Naval Air Systems Command, Washington, DC — 23 Apr 79
12. REPORT DATE: April 23, 1979
13. NUMBER OF PAGES: 58

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) — 59 pp
15. SECURITY CLASS. (of this report): UNCLASSIFIED
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited. — NRL-MR-3980

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Multiplexor	Microprocessor	Data acquisition
NTDS Channel	8050	Time sharing
RS232	Interface	Programmable interface
AN/UYK-20	Computer	

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A multiport RS232-NTDS Multiplexor will be discussed. This microprocessor driven hardware device has the capability of multiplexing up to 48 RS232 ports into a single parallel NTDS channel obeying intercomputer protocol. Each RS232 port has software programmable functions similar to AN/UYK-20 RS232 functions. The device is modular so that RS232 ports may be added incrementally as needed. The device is flexible: a very general priority interrupt structure is provided by a combination hardware logic and microprocessor firmware. As a result of this flexibility and and modularity, the device can easily accommodate special purpose peripheral devices. (Continues)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

 251 950

20. Abstract (Continued)

The motivation for construction of the device will be discussed along with possible applications. Salient design features affecting flexibility, modularity and throughput will be discussed.

This is a final report on this phase of the problem.

TABLE OF CONTENTS

1.0 Background 1
1.0.0 General Description 1
1.0.1 NTDS Parallel Channels 2
1.0.2 RS232 Serial Channels 2

2.0 RS232-NTDS Multiplexor 3
2.1 RS232--A Second Look 3
2.2 AN/UYK-20 Hardware Configuration 4

3.0 Design Philosophy of the NTDS-RS232 Multiplexor 5
3.0.0 Description of the Multiplexor 5
3.0.1 Control Philosophy 8
3.0.2 Control Structure 9

4.0 Hardware 10
4.1 Handshake with Multiplexor 10
4.2 Multiplexor Hardware Operation 11
4.2.0 Port to AN/UYK-20 11
4.2.1 AN/UYK-20 to Port 11

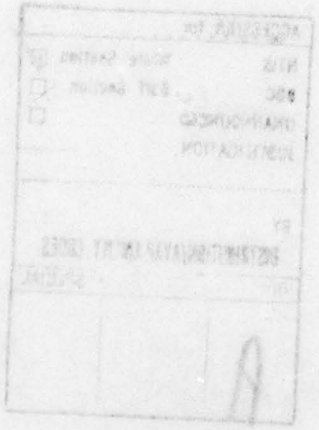
5.0 Software 13
5.1 AN/UYK-20 Software 13
5.1.0 Packet Format 13
5.1.1 Function Requests 14
5.1.1.0 STAT 15
5.1.1.1 Channel Functions 15
5.1.1.2 Port Functions 15
5.1.2 Status 16
5.1.3 Summary 16
5.2 8080 Microprocessor Software 17

6.0 Design and Debug 18
6.1 8080 Design and Debug 18

7.0 Conclusion 19

FIGURES

1.0	AN/UYK-20 Block Diagram	1
3.0.1	Block Diagram of Multiplexor	6
3.0.2	Functional Representation of Multiplexor	7
3.0.3	Data Word Format for AN/UYK-20 to Multiplexor Interface	8
5.1	MX$$ Control Packet Format	14



APPENDICES

A. Functional Logic Diagram of Multiplexor 21

B. Assembler Listing of MX$$ and Test Program .. 27

C. Flowchart for MX$$ 39

D. PL/M Driver for Multiplexor 46

ABSTRACT

A multiport RS232-NTDS multiplexor will be discussed. This microprocessor driven hardware device has the capability of multiplexing up to 48 RS232 ports into a single parallel NTDS channel obeying intercomputer protocol. Each RS232 port has software programmable functions similar to those for AN/UYK-20 RS232 ports. The devices is modular so that RS232 ports may be added incrementally as needed. The device is flexible: a general priority interrupt structure is provided through a combination of hardware logic and microprocessor firmware.

The motivation for construction of the device will be discussed. Salient design features affecting flexibility, modularity and throughput will be discussed.

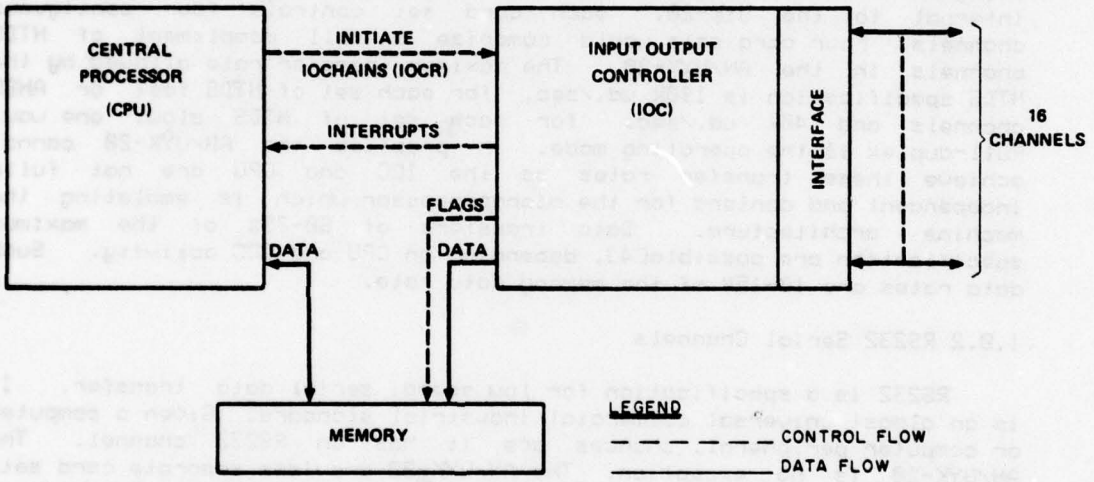
This is the final report on this phase of the problem.

1.0 Background

The AN/UYK-20 is the standard Navy minicomputer for surface applications. It is a completely militarized, high performance processor with extensive input-output capabilities. This report is concerned with an interface designed for enhanced utilization of these input-output capabilities.

1.0.0 General Description

In addition to the central processor(CPU), the AN/UYK-20 architecture provides an Input-Output Controller (IOC) for handling data and control transfers with external devices via one of the input-output channels[Fig. 1.0]. The IOC may execute sequences of special input-output instructions known as 'chains' to control any of the processor's 16 channels. As the number of input-output instructions is quite extensive, channel control is very flexible.



a) INTERFACE TYPES: NTDS PARALLEL (SLOW, FAST, ANEW), RS232, NTDS SERIAL, MIL-STD-188C

Figure 1.0 AN/UYK-20 Block Diagram

The channels themselves follow one of three types of handshaking conventions for data transfer: NTDS, MIL. STD. 188C or RS232. 'NTDS' refers to 'Naval Tactical Data Systems' interface described in MIL. STD. 1397[1]. There are three types of parallel NTDS interfaces and one serial type. RS232[2] is a low speed serial protocol especially designed to interface with data communications equipment(eg., modems). RS232 is an industrial standard specified by the 'Electronics Industries Association'. MIL. STD. 188C is a serial military specification for a low speed interface resembling RS232.

Note: Manuscript submitted March 1, 1979.

Each channnel type may be further specified as to speed, voltage levels, and transfer modes. Thus the serial RS232 channel may be synchronous or asynchronous, of various baud rates, etc. The NTDS channels may be parallel(slow , fast, or ANEW) or serial[3].

The address(0-15) of each particular input-output interface type in the AN/UYK-20's 16 channels is somewhat flexible and determined by the installation. Each channel is provided with two 120-pin connectors to the outside world. Therefore, sufficient connector hardware is supplied on the processor to support 16 parallel, full-duplex channels .

The remainder of this report shall be concerned with NTDS parallel and RS232 serial channels. Many RS232 ports will be interfaced to an AN/UYK-20 NTDS parallel channel by a device to be described.

1.0.1 NTDS Parallel Channels

The three type of NTDS parallel channels(slow,fast, ANEW) are identical as far as the programmer is concerned. The NTDS slow channel differs from the two fast channels in transfer rate, timing and voltage levels. The NTDS fast and ANEW channels differ only in voltage levels assigned for the one and zero states. All channel logic is on card sets internal to the UYK-20. each card set controls four contiguous channels. Four card sets would comprise a full complement of NTDS channels in the AN/UYK-20. The maximum transfer rate allowed by the NTDS specification is 190K wd./sec. for each set of NTDS fast or ANEW channels and 40K wd./sec. for each set of NTDS slow, one way. Full-duplex is the operating mode. In practice the AN/UYK-20 cannot achieve these transfer rates as the IOC and CPU are not fully independent and contend for the microprocessor which is emulating the machine architecture. Data transfers of 60-75% of the maximum specification are possible[4], depending on CPU and IOC activity. Such data rates are 10-15% of the memory data rate.

1.0.2 RS232 Serial Channels

RS232 is a specification for low speed, serial data transfer. It is an almost universal commercial industrial standard. Given a computer or computer peripheral, chances are it has an RS232 channel. The AN/UYK-20 is no exception. The AN/UYK-20 provides separate card sets for synchronous (clock triggered baud sampling) and asynchronous (data triggered baud sampling) RS232 channels. Each RS232 card set or provides two channels. The asynchronous channels transfer at one of four programmable rates; the maximum rate is 2400 baud. The synchronous channels can transfer 0-9600 baud., depending on the sampling clock. Since the RS232 channels are serial channels, each one uses considerably fewer than half the 480 IO connector pins (ie. four 120-pin connectors) available for the two channels that are used by each RS232 card set.

2.0 RS232-NTDS Multiplexor

Section 1 above provided an overview of AN/UYK-20 IO properties. The remainder of this report will focus on the description of a device that marries NTDS and RS232 channels external to the AN/UYK-20. The heart of the device is a microprocessor which controls data switching and buffering to implement the function of multiplexing many RS232 channels into a single NTDS channel. Thus, instead of using multiple AN/UYK-20 RS232 channels, only a single NTDS channel is needed. This channel is connected through the multiplexor to many RS232 devices. One need not utilize RS232 card sets, instead, an NTDS channel is tied to the external multiplexor. Firmware in the multiplexor and an AN/UYK-20 software routine allow program control of the multiplexor.

2.1 RS232--A Second Look

Serial transmission on RS232 channels occurs commonly at rates up to 19.2 kilobaud asynchronous(eg. CRT terminals), and up to 50 kilobaud synchronous(eg. remote data terminals). Transfer rates beyond this exceed the RS232 specification which limits slew rate and does not provide for a balanced transmission line.

The motivation for the RS232-NTDS multiplexor is as follows:

a) The RS232 interface is widely available in commercial peripheral equipment at little or no extra cost. On the other hand peripherals with NTDS interfaces demand premium prices suggesting the use of the multiplexor as an interface between a commercial peripheral with RS232 channels and an AN/UYK-20 NTDS channel.

b) The RS232 protocol was designed for interface to a modem allowing remote communications over, say, phone lines. The multiplexor, having the potential for many more RS232 channels than the AN/UYK-20 mainfame, provides an ideal means for interfacing the AN/UYK-20 to low speed serial data lines from many remote location (eg. sensor inputs or timesharing user terminals).

c) The extreme slowness of the RS232 channels compared to the NTDS channel should be noted. While a transfer rate of 100-150 kilowords/sec.(1.6-2.4 megabits/sec.) is possible with NTDS channels, RS232 chanels are limited to 2400 baud/sec. asynchronous and 9600 baud/sec. synchronous. The multiplexor was designed to ameliorate the mismatch between the potential AN/UYK-20 IO channel bandwidth and RS232 data rates.

d) Activating the serial RS232 interface in the AN/UYK-20 requires IO instructions apart from those needed to implement io transfers in NTDS channels. Externalizing RS232 channels reduces the set of instructions needed for input-output operations and makes possible standardizing AN/UYK-20 interfaces to a single type: NTDS parallel.

2.2. AN/UYK-20 Hardware Configuration

The number of card slots required by each set of four NTDS channels is equal to the number of slots required by four RS232 channels. However, a card set containing two RS232 channels may preclude usage of a set of four NTDS channels. The implication is a devastating trade-off in memory bandwidth and connector pins for every set of RS232 channels in the machine. That is, 120 IO pins and mating connector are devoted to each RS232 channel requiring less than 25 lines. Each channel connector port dedicated to RS232 protocol has a bandwidth of less than 10 kilobit/sec. While the potential bandwidth of the channel is in excess of 1 megabit/sec.

3.0 Design Philosophy of the NTDS-RS232 Multiplexor

The philosophy behind the multiplexor is essentially this: externalize all RS232 channels in the multiplexor. Configure the AN/UYK-20 solely with NTDS parallel interfaces. Thus, only one NTDS channel need be dedicated to service all RS232 ports. Only two AN/UYK-20 IO connectors need be dedicated to service all RS232 ports. A subset of AN/UYK-20 IO instructions is then sufficient for all input-output control.

The benefits to be gained from this approach are:

a) Minimize/utilize effectively interconnection hardware.
b) Maintain the high IO bandwidth possible with an NTDS parallel interface.
c) Provide many RS232 ports.
d) Provide a flexible interface.

In addition, the benefits of using microprocessor control will be emphasized as the design is discussed.

3.0.0 Description of the Multiplexor

Figure 3.0.1 shows a block diagram of the multiplexor. The AN/UYK-20 communicates with the multiplexor via an NTDS channel. The multiplexor contains an NTDS interface which is implemented with a combination of hardware and software. Control of the system is provided by an 8080 microprocessor, an 8-bit, programmable NMOS integrated circuit[5].

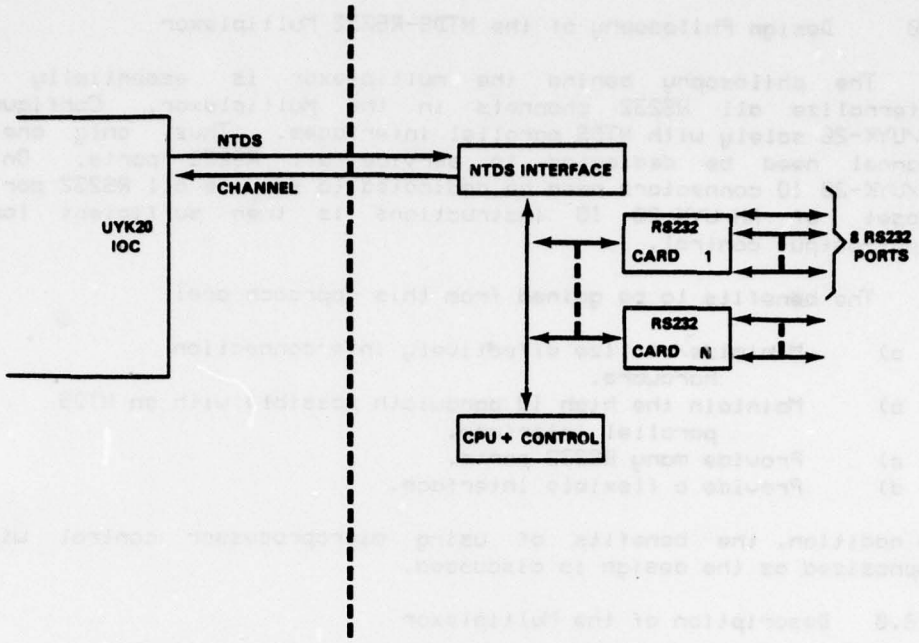


Figure 3.0.1 Block Diagram of Multiplexor

The multiplexor has RS232 ports grouped on cards which may be added to the system in a modular fashion. Each card contains up to eight USARTs(Universal Synchronous/Asynchronous Receiver/ Transmitter) interfaced to the external world using RS232 protocol.

Each USART is programmable as to baud rate and mode of operation(synchronous/asynchronous). Certain RS232 control lines may be set or cleared under program control . The ability to set port mode is not available in internal AN/UYK-20 RS232 ports. Also, the maximum baud rate in either mode is significantly greater in the multiplexor USARTs than in the internal AN/UYK-20 RS232 ports. These improvements are consequences of more recent microprocessor technology.

Figure 3.0.2 shows a functional diagram of the multiplexor. The microprocessor functions as a switch and decoder and may read or write any memory location in the multiplexor under software control. Memory local to the multiplexor consists of random access memory(RAM), read only memory(ROM), locations associated with RS232 data and control and locations associated with NTDS data and control. Thus all devices in the multiplexor are treated as memory locations. RAM provides scratchpad and buffer storage. Software that defines multiplexor operation(in conjunction with the intrinsic hardware) is stored in ROM.

Data to and from the multiplexor is in a word format, each word being 16 bits. Each word contains a byte of address/command information(upper eight bits) and a byte of data(lower eight bits)[Fig. 3.0.3]. The address byte is further subdivided into an intracard address(lowest 3 bits), a card address(next 3 bits), a command bit(bit

6) and a spare bit(bit 7). The intracard address identifies the particular port on the card specified by the card address. The commmand bit separates data from command/status information.

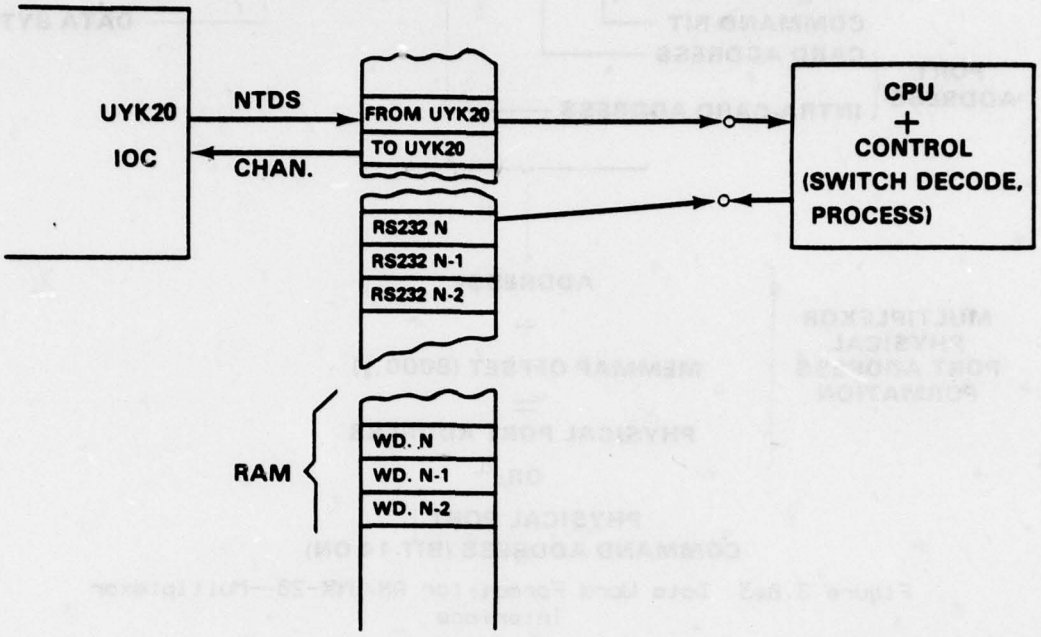


Figure 3.0.2 Functional Representation of Multiplexor

All data transfers over the NTDS channel follow NTDS intercomputer protocol which renders the multiplexor-AN/UYK-20 interface symmetrical; neither device is intrinsically a master. The need for command transfers(NTDS forced command mode[6]) is obviated since command information is contained in the data word.

All input output ports in the multiplexor look like memory. A typical operation of the multiplexor would be to have the microprocessor read the AN/UYK-20 memory locations, decode the port or command address then write the appropriate RS232 memory location.

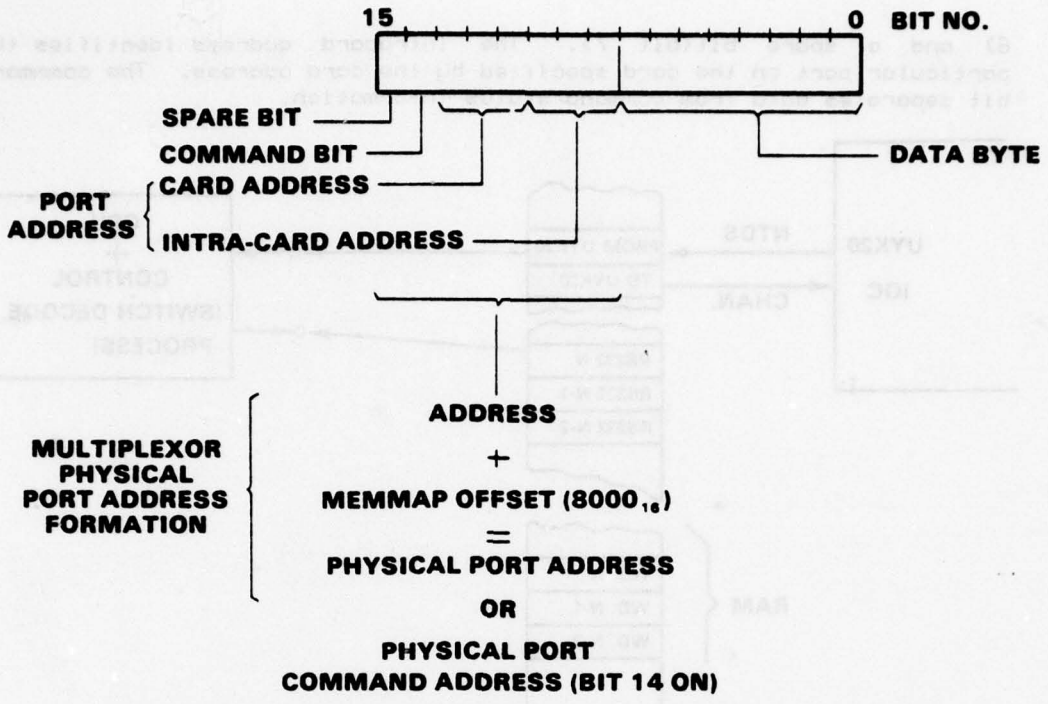


Figure 3.0.3 Data Word Format for AN/UYK-20--Multiplexor Interface

3.0.1 Control Philosophy

When the multiplexor is not being taxed, eg. when a few 300 baud terminals are being used, the type of control structure used to control data flow is moot: any scheme implementable in software should suffice. If the multiplexor is exercised more strenuously, then it is well to consider the possible control structures and pick the one best suited to the application. Different applications warrant different control structures. For instance, suppose the ports connect many low speed terminals to the AN/UYK-20. In this case, service requests would occur infrequently and randomly. If a single port were to become active in an otherwise quiescent system, using an interrupt structure would guarantee a fixed time to service given by the hardware interrupt service timing. This would be preferable to polling in software which would give a variable time to service of average duration tpoll*n/2, where 'n' is the number of ports and 'tpoll' is the time to poll a single device.

On the other hand, suppose the system is busy, i.e. several ports handling data work more or less continuously; suppose also tpoll is less than the interrupt service time. In such a case polling would be preferable. If the ports have some priority assigned, using the prioritized interrupt structure available is, perhaps, best. Though in some sense sense we could assign priorities in a polling situation by 'super commutation', that is, if a,b,c,d... are ports one polls 'abacad...' or any sequence where polling does not occur in ordinal order.

There is yet another type of control structure to consider, namely buffered transfer; this type of interrupt gives the device being serviced priority. Since RS232 ports are rather slow, it is unlikely they would be used in such a mode. The AN/UYK-20 NTDS interface is quite another matter, however. The speed of the transfer may well warrant buffered transfer to cut down on software and hardware overhead involved in recognizing and servicing a system interrupt. Random access memory in the system and the flexible interrupt scheme make this mode implementable in software.

In summary, in a sparse, random request environment use a prioritized interrupt structure; in a busy, uniform environment, use polling.

3.0.2 Control Structure

As the design of the prototype multiplexor progressed, it became obvious that in this type of controller application, the utility of the software was determined by the quantity of controllable hardware in the multiplexor. Though believed fervantly, this concept could not always be followed. The control structure options, however, obey the concept. Minor modifications allow different modes of operation:

Each port may be polled to see if it is active.
Each card may be an interrupt.
Each port may be an interrupt (6 port maximum).
Each card may be polled.
Each card may be swept of all pending intrrrupts.

The interrupt structure is prioritized and the interrupt address is stored in a memory location. The 8080 hardware vector structure may also be used. The intracard address of each port on the card is also available, so that a 6 bit interrupt address exists.

With additional hardware, the 8080 may support eight vectored interrupts(0-7). The multiplexor supplies the extra hardware needed so that eight separate interrupts, each having a unique interrupt service routine are available. Vector interrupt 0 is the system reset, 1 is the NTDS interrupt, 2-7 are available for RS232 cards. RS232 cards have highest system priority. In addition, as the USARTs are programmable and may be queried for status, system polling is possible.

A choice must be made among possible interrupt schemes and the multiplexor must be hard-wired accordingly. The multiplexor configuration used with the software described in this report was based on a card interrupt priority structure. However,the six bit port address as read from the interrupt latch provided address information.

4.0 Hardware

In order not to 'reinvent the wheel' commercially available microprocessor CPU and RAM/ROM memory cards were utilized. This choice greatly facilitated the multiplexor design. Numerous packaging options of this type are available. Judicious choice of pre-packaged aids should be a first step in any new microprocessor design.

It was decided to package up to eight RS232 ports on a card which could be replicated to provide up to the maximum number of ports. This choice arose naturally from the card size and interrupt circuitry. Other cards provide CPU, control and decode, memory and the NTDS interface[Fig. 3.0.1].

As command information is contained within the data word, simple data transfer in intercomputer mode was adequate. One consequence of the choice of NTDS-intercomputer mode is that the data from the UYK-20 is held on the channel lines until the 8080 can respond. Similarly, the 8080 microprocessor will hold its data on the channel data lines until the AN/UYK-20 responds. The 8030 bus has nowhere near the IO bandwidth or speed of an AN/UYK-20 NTDS fast channel, but the microprocessor could keep an NTDS slow channel fairly active.

4.1 Handshake with Multiplexor

Handshake protocol follows that described for NTDS parallel intercomputer data transfers[7]. The hardware sequence from the AN/UYK-20 is as follows:

a) UYK-20 puts data on lines, sets READY.
b) Multiplexor latches data, generates interrupt.
c) 8080 recognizes interrupt according to appropriate priority and vectors to interrupt service routine(ISR).
d) Software control in ISR reads lower and upper byte of multiplexor buffer latch.

Reading the upper byte of the buffer latch causes the RESUME signal to be generated by the multiplexor. The RESUME signal clears the READY signal. When READY is cleared the multiplexor data latches are released. The AN/UYK-20 may then output another data word.

The hardware sequence of events for data transfer to the AN/UYK-20 follows:

a) The 8080 writes an output buffer latch, READY signal is generated.
b) The AN/UYK-20 senses READY and samples data lines.
c) The AN/UYK-20 generates RESUME which clears buffer latch and READY. The multiplexor may then output another data word.

A consequence of the hardware handshake sequence is that data is valid in the buffer latches only until handshake completion, ie. only while READY line is high. Software in the multiplexor must take this fact into account and not use the latches as permanent storage locations. Under software control, the READY line may be sampled to prevent

overwrite on output to AN/UYK-20 or to monitor input to the multiplexor in a polled environment.

A logic block diagram of the multiplexor is included in APPENDIX A.

4.2 Description of Multiplexor Hardware Operation

4.2.0 Port to AN/UYK-20

Each port is a type 8251 Universal Synchronous/ Asynchronous Receiver/Transmitter(USART)[8]. The ports are grouped on the RS232 cards, a maximum of 8 per card. In the current implementation, each card requires one interrupt vector. Six unique vectors are available, two others are used for AN/UYK-20 interface and system restart.

When a port receives a character, a control line(RXRDY) is raised which is used as an interrupt. Only a read of or command to the port will reset this line. This does not preclude the port being overwritten.

Cards have a daisy-chain priority: if a card has no port requests pending, the next card in the chain is enabled. If a card is enabled and a port on the card receives a character, the intra-card priority is resolved, and port address will be broadcast on a 3-bit bus. The card generates an interrupt pulse which is logged into the master interrupt latch. The interrupt pulse will be generated each time interrupt priority is resolved, if the card is enabled. This sequence of events will occur until the request is serviced and the port is read. When the card has highest priority, an interrupt is generated to the 8080 which responds with interrupt-acknowledge(INTA). At this point, system hardware vectors the microprocessor to the appropriate service routine and the interrupt address is further decoded under software control.

A 16-bit word consisting of an upper byte of address and control information and a lower byte of data is transferred to the AN/UYK-20 under software control.

Since the USART port status, including the interrupt bit, RXRDY, is software addressable, one may use a software polling scheme instead of priority interrupt scheme to drive the multiplexor. If the interrupt structure is turned off, the status of the AN/UYK-20 handshake lines must also be polled under software control. A software polling scheme is not currently used in the multiplexor.

4.2.1 AN/UYK-20 to Port

The AN/UYK-20 will write the multiplexor data latches and set the READY line. The setting of the READY line generates an interrupt to the microprocessor. The AN/UYK-20 interface has lower priority than any RS232 card in the system. Reading of the data latches by the 8080 will generate a RESUME signal. The port address is decoded from the data under software control, and the port memory address is generated. The data is decoded and transferred to the port.

Since the READY signals from the AN/UYK-20 and the multiplexor are monitored in a hardware status latch, it is possible to poll the AN/UYK-20 or prevent an output overwrite from the multiplexor to the UYK-20. Monitoring of READY signal is under software control.

5.0 Software

5.1 AN/UYK-20 Software

The multiplexor is designed to be controlled from the AN/UYK-20. Sixteen-bit(word) data transfers contain data byte, port address and command information. The programmer is partially isolated from the multiplexor hardware by an AN/UYK-20 assembly language program called MXSS. MXSS is compatible with the AN/UYK-20 support software LEVEL1/LEVEL2 format for IO handlers[9] in that it utilizes a packet of five words to control transfers between the AN/UYK-20 and the multiplexor[Fig. 5.1]. The packet format is similar to LEVEL1/LEVEL2 format for packets. The standard call is:

```
JLRR    R15,MXSS        ;JUMP TO SUBROUTINE, SAVE RETURN
                        ;ADDRESS IN R15.
+       PKTADDRESS      ;ADDRESS OF PACKET GOES HERE.
```

MXSS allows one data input and one data output on the channel to the multiplexor to run concurrently. If the channel is reading and another read function is requested, a busy status will be returned to the user via the packet status byte and the second request will not be honored. Channel write functions are treated in like manner.

5.1.0 Packet Format

The five word packet format is shown in Figure 5.1. The upper byte of word zero is reserved for status of the current request as will be described. The lower byte of word zero specifies the type of action request by the packet, ie. the function request.

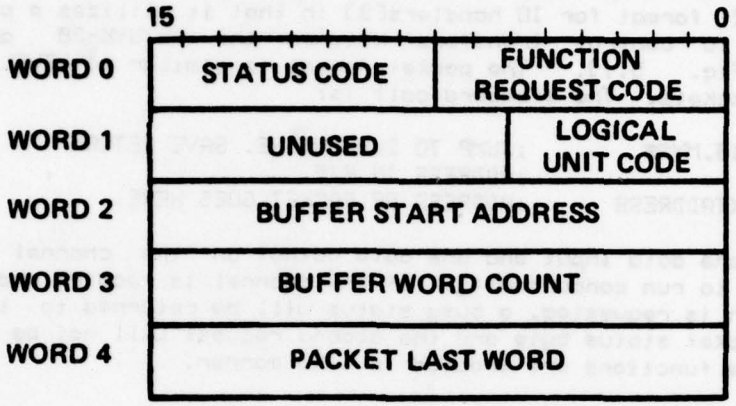


Figure 5.1 MXSS Control Packet Format

The lower six bits of packet word one contain the logical unit(LU) code. This code is a pointer to a table called 'LUMAP'. The table entries in LUMAP are the physical addresses of the RS232 ports in the particular multiplexor configuration. Special entries are made in LUMAP for idle or non-existent ports. The physical address is formed by shifting the port interrupt address to the upper byte of a word. LUMAP must be initialized with the proper physical port addresses if MXSS is to control the multiplexor properly. The initialization depends on the ports existing in the particular AN/UYK-20-multiplexor configuration. All existent ports should have their interrupt addresses entered into the upper byte of the appropriate location in LUMAP. Ports should be initialized in the idle state, ie. signbit of the LUMAP entry set to '1'. Octal '1000000' is entered for non-existent ports.

Packet word two gives the starting address of a data buffer; packet word three gives the buffer size. The buffer size is limited to 4096 words.

Packet word four is used by MXSS to return mode and command information on the USART port designated by the packet LU code. A table called 'MCMAP' keeps an updated record of the last mode and command instruction issued to each USART in the system.

5.1.1 Function Requests

Function requests are coded in the lower eight bits of packet word 0. The function requests currently implemented in MXSS are as follows:

FUNCTION	REQUEST CODE(OCTAL)
READ	000
WRITE	001
INIT	002
TERM	006
STAT	011
CMD	012
READIM	013
WRITIM	014

Care was taken to be compatible with standard LEVEL1/LEVEL2 software conventions for IO packets.

Functions may be classified into three groups:

STATUS	STAT
CHANNEL FUNCTIONS	TERM,READIM,WRITIM
PORT FUNCTIONS	INIT,CMD,WRITE,READ

5.1.1.0 STAT

'STAT' does not access the channel but rather gets multiplexor status from tables stored in the AN/UYK-20 memory and updated by MXSS. Port status(active,idle,non-existent) and the last mode and command instructions issued may be obtained by a call to STAT. The mode/command information is returned in packet word 4.

5.1.1.1 Channel Functions

Channel functions are independent of port assignment.

'TERM' aborts all multiplexor channel activity and resets all ports regardless of status or choice of logical unit(LU). Port status is set to idle for all ports.

'READIM' turns on channel to listen for input from any port. It is the programmer's responsibility to decode address information in a completed READIM buffer to ascertain which ports have input information.

'WRITIM' outputs a data buffer without modification. It is the programmer's responsibility to encode address information in each buffer word prior to activating a WRITIM buffer output. Consequently, the programmer may choose any sequence of port outputs. For example, an output sequence maximizing the time between outputs to a given port would minimize system delay owing to waiting for the port to complete output. It is possible to issue commands using WRITIM, but this practice is to be avoided as system status may very easily be lost.

5.1.1.2 Port Functions

The packet logical unit code points to an entry in LUMAP which is the physical port address of a port. Recall that the physical port address depends on the interrupt address of the port and that this address is placed in the upper byte of the appropriate LUMAP location. Of course, LUMAP may 'map' any logical unit to any physical port

address. MXSS as shown in APPENDIX B is configured for 32 ports, ie. LUMAP is 32 words long.

'WRITE' will write data to a given LU. The data is in the lower byte of each buffer word; MXSS will fill in the physical port address for the given LU.

'READ' is similar to 'READIM',since any port may input data to the AN/UYK-20. Read differs from READIM in that a check on port status is performed. If the port is idle or non-existent, the read request will not be honored and the appropriate status(idle or nonexistent) will be returned in the packet status byte.

'INIT' initializes a given LU. The port is reset and a mode instruction followed by a sequence of commands is output to the port. The MCMAP in MXSS records the last mode and command issued to the port when INIT terminates successfully. As the command update is derived from packet word 4, it is the programmer's responsibility to supply the correct command information(possibly 0 if only a mode instruction is issued). A reset command in an INIT buffer will abort the request and return a data error status.

'CMD' will output a string of commands to a given LU. The command status is updated using packet word 4. Proper command status is again the programmmer's responsibility, and the last command must be supplied in packet word four. A reset issued in a CMD buffer will reset the port. MXSS will supply the physical port address for each buffer word for CMD and INIT requests.

5.1.2 Status

There are seven status codes which are 'orred' into the packet status byte as follows:

```
STATUS          CODE

FCN COMPLETE    000000  ;PACKET REQUEST COMPLETE.
IDLE            001000  ;PORT IS IDLE.
BUSY            013000  ;ANOTHER CHANNEL READ/WRITE
                        ;REQUEST IS BEING HONORED.
DATAERR         040000  ;DATA ERROR.
FRNV            040400  ;FCN. REQUEST CODE INVALID.
NODEV           041000  ;NO PHYSICAL DEVICE
                        ;CORRESPONDS TO THIS LU.
IOACTIV         177400  ;PACKET REQUEST IS BEING
                        ;SERVICED.
```

5.1.3 Summary

MXSS provides the AN/UYK-20 user with control of the multiplexor. MXSS is seen as a kernal input/output handler for the AN/UYK-20-multiplexor system upon which more software may be developed. Such software could further isolate the user from the internal workings of the multiplexor, eg. command formats for the USARTs. A listing of MXSS and program flowchart are provided in the appendices.

5.2 8080 Microprocessor Software

In the current implementation of the multiplexor, 8080 software has been kept as simple as possible. Software was written in a high level language cross-compiler.

A driver program for the multiplexor is given in APPENDIX D the program's basic function is address decoding. Words from the UYK-20 are split into address and data bytes. The memory map port, or port command, address is formed and the data byte is transferred to the port.

In transfer to the UYK-20, the interrupt address is used to form the port address. The port is read, and the address information is concatenated to the data byte to form a data word. The UYK-20 is then written.

The particular program shown in APPENDIX D activates three ports. one port (port2) is activated and a local sign-on message is output to the port by the multiplexor. The other two ports are put in command mode. .The AN/UYK-20 software may thus assume a non-ambiguous state (command) for the ports.

As applications of the multiplexor are fixed, 8080 software may be added to perform more functions locally as needed. examples of such functions are: local character delete, operation in line mode, ASCII to other code conversion, data buffer transfer, parity checks, etc.

6.0 Design and Debug

The multiplexor is a software-hardware device. With the advent of microprocessor technology, such multifaceted designs will become more and more common. Microprocessor software is deliberately introduced into the design: 8080 software controls and complements the multiplexor hardware. The added burden of software in the design is compensated for by the flexibility in the instrument so designed. This is especially true of a device like the multiplexor where many potential interconnect schemes, modes of operation and pre-processing functions could be conceived.

The interaction of hardware and software in the design must be cut at some point so the design may be firmed up. This interaction occurs again in the debug phase. There were 8080 software problems thought to be multiplexor hardware problems, AN/UYK-20 hardware problems thought to be multiplexor hardware problems and AN/UYK-20 software problems thought to be caused by problems in the multiplexor. Sufficient richness exists to keep the design engineer entertained!

6.1 8080 Design and Debug

An effort was made to keep 8080 software simple. At first, 8080 assembly language was used, and programs were always found to be under 256 bytes. Since 1024 bytes of PROM storage were available, it was decided the final software package would be written in PL/M, INTEL Corporation's high level language for the 8080[10]. A tradeoff was made: the less efficient use of memory could be tolerated, while gaining the benefits in documentation and ease of program modification provided by the high level language. The PL/M program in APPENDIX D required 363 bytes of ROM storage.

The value of programmability can best be shown by this example: At one point in the debug, one dead bit, was discovered in the AN/UYK-20 NTDS driver and one in the receiver. No replacement card was available. Using a parity checking routine in the 8080, this problem was corrected and the debug process could be continued.

It was found that 8080 software modification was not frequent; consequently, the use of a cross-assembler and a cross-compiler on a PDP-10 timesharing system to generate papertape object code input to a PROM programmer proved a satisfactory way to develop multiplexor software.

At any stage, support hardware was traced using a logic analyzer, a debug aid that proved invaluable.

7.0 Conclusion

The RS232-NTDS multiplexor should prove a useful tool in those cases where serial data is to be input to an AN/UYK-20 constrained to have NTDS channels. The constraint may be by fiat or from other considerations such as:

a) Reducing card types in the machine
b) Reducing external cabling and hardware
c) Preserving NTDS channel bandwidth

The multiplexor should be a useful tool where rapid or temporary connection of commercial peripherals is desired, say in software development. Premium cost militarized peripherals or peripherals with NTDS interfaces can thus be avoided. It should prove useful where many signals are monitored as in timesharing or remote data gathering. It provides a means of accessing serial data links for phone line or modem communication. The basic philosophy behind the design is that there are cases where decentralizing control is beneficial. Microprocessor technology has made such decentralization very tempting!

This exposition has tried to include some of the tradeoffs to be made in the microprocessor design, and to provide suggestions as to a reasonable design/debug procedure.

It is hoped the multiplexor design effort will serve as a background for further work.

REFERENCES

1. 'Input/Output Interfaces, Standard Digital Data, Navy Systems', MIL-STD-1397(SHIPS), Dept. of the Navy, Naval Ship Systems Command, Washington, D. C., 30 August 1973.

2. 'Interface Between Data Terminal Equipment and Data communications Equipment Employing Serial Binary Data Interchange', EIA STD. RS232c, Electronic Industries Association, Washingtion, D. C., August 1969.

3. MIL-STD-1397, page 1.

4. Robert Welsh, 'AN/UYK-20 I/O', AN/UYK-20 User's Conference, 30 Nov. to 2 Dec. 1976, San Diego, Calif., Talk Given 1 December 1976.

5. 'INTEL 8080 Microcomputer Systems User's Manual', INTEL Corp., September 1975, Chapters 2 to 4 and pp. 5-13 to 5-20.

6. MIL-STD-1397, page 9.

7. MIL-STD-1397, page 9.

8. '8080 Microcomputer Systems User's Manual', pp. 5-135 to 5-146.

9. 'User's Handbook for AN/UYK-20(v) Computer', Vol. 1, Sperry Univac, St. Paul, Minn. for U. S. Navy, Naval Electronics Systems Command(Contract N00039-74-6-0049), May 1976, pages 2-3-1 ff. and 3-3-1 ff..

10. '8008 and 8080 PL/M Programming Manual', Revision A, INTEL Corp., 1975.

11. '8080 PL/M Compiler Operator's Manual', Revision A, INTEL Corp., 1975.

12. 'User's Handbook For AN/UYK-20(v) Computer', Vol. 3, Part 6.

13. '8080 Microcomputer Systems User's Manual', Chap. 4.

APPENDIX A

FUNCTIONAL LOGIC DIAGRAM OF THE MULTIPLEXOR

GLOSSARY FOR BLOCK DIAGRAM

Note that overbar signifies an inverted signal; 1=+5 volts=true.

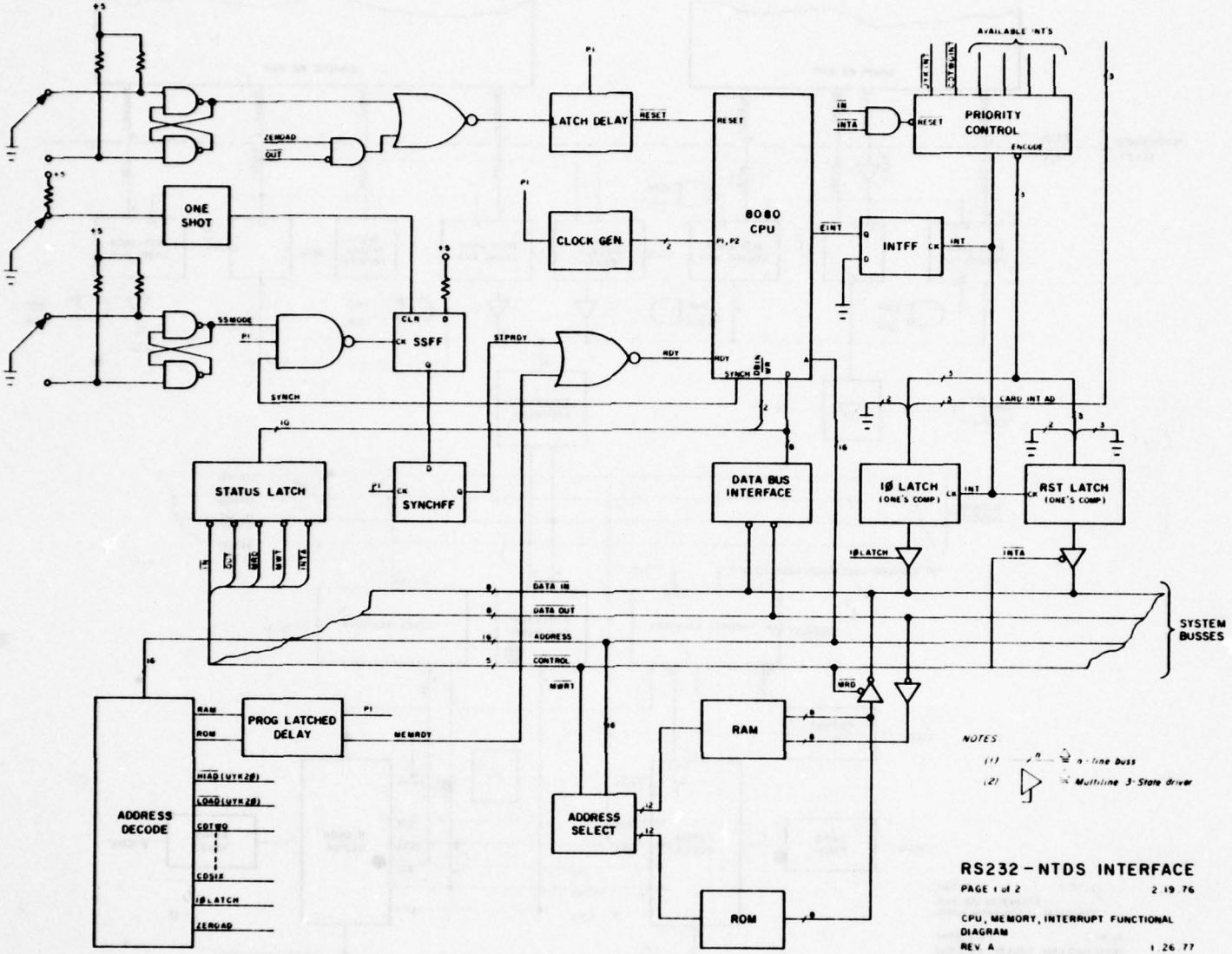
SIGNAL	DESCRIPTION
ADDRESS0-15	16-bit 8080 address bus
CDFIVE	Card enable for card five
CDFOUR	Card enable for card four
CDTHREE	Card enable for card three
CDTWO	Card enable for card two
CDTWOINT	Interrupt for RS232 card 'CDTWO'
DATAIN0-7	8-bit data bus for input to 8080
DATAOUT0-7	8-bit data bus for output from 8080
DBUS0-7	RS232 card internal bus
EINT	Latch external interrupt
FMUYK0-15	16-bit UYK-20 output --NTDS levels
HIAD	Address of high byte of UYK-20 latches
IN	8080 input flag
IN0-15	16-bit input from UYK-20 --TTL levels
INT	External interrupt pulse
INTA	8080 interrupt acknowledge in response to interrupt request
IOLATCH	Latch to store 6-bit system interrupt vector
IR0-7	RS232 card interrupt lines

GLOSSARY FOR BLOCK DIAGRAM(CONT.)

SIGNAL	DESCRIPTION
LOAD	Address of low byte UYK-20 latches
MEMORY	Memory-ready line to synchronize slow memory with 8080
MRD	8080 read pulse
MWRT	8080 write pulse
ON0-7	Selects one of 8 RS232 ports on serial interface card
OUT	8080 output flag
OUT0-15	Output to UYK-20 --TTL levels
P1	8080 clock phase 1
P2	8080 clock phase 2
RAM	Random access memory
RDY	8080 'ready" line
RDYCPU	'Ready' from multiplexor--NTDS levels
RDYOUT	'Ready' from UYK-20 --TTL levels
RDYUYK	'Ready' from UYK-20 --NTDS levels
RDY80	'Ready' from multiplexor --TTL
RESET	System reset--resets 8080 also
RESIN	'Resume' from UYK-20 --TTL levels
RES	'Resume' from multiplexor --NTDS levels
RESUYK	'Resume' from UYK-20 --NTDS levels
RES80	'Resume' from multiplexor --TTL levels

GLOSSARY FOR BLOCK DIAGRAM(CONT.)

SIGNAL	DESCRIPTION
ROM	Read only memory
RSCK	Crystal source clock for RS232
RSCK0	RSCK/N for even ports
RSCK1	RSCK/(2*N) for odd ports
SINGLE STEP	Front panel single step pulse
SSMODE	Front panel single step enable
STATAD	Line to enable UYK-20 status
STPRDY	Pulse enabling next instruction when in single step mode
SYNCH	8080 'SYNCH' signal
TOUYK0-15	16-bit UYK-20 input --NTDS levels
UYKINT	Interrupt from UYK-20 interface card
ZEROAD	Line indicating address 0000



RS232 – NTDS INTERFACE

PAGE 1 of 2 2 19 76

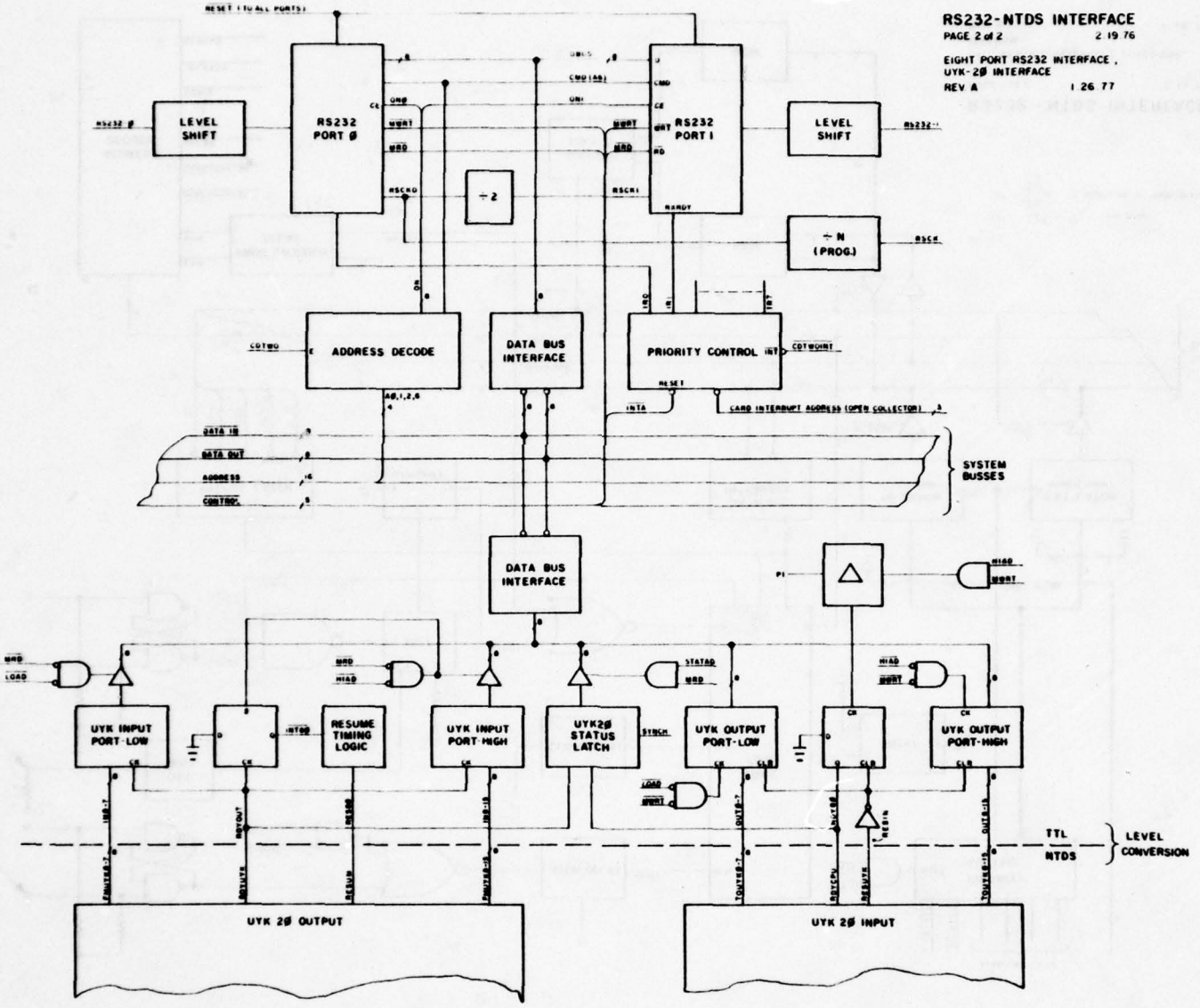
CPU, MEMORY, INTERRUPT FUNCTIONAL DIAGRAM

REV A 1 26 77

RS232-NTDS INTERFACE

PAGE 2 of 2 2 19 76

EIGHT PORT RS232 INTERFACE ,
UYK-2Ø INTERFACE
REV A 1 26 77



APPENDIX B

ASSEMBLER LISTING OF MXSS AND TEST PROGRAM

The assembly language used is not standard AN/UYK-20 'ULTRA'[11]. To translate to ULTRA, apply the following rules:

For RK instructions, add suffix 'K' to normal ULTRA mneumonic.

For RX instructions, delete asterisk from normal ULTRA mneumonic.

The assembled code is shown to resolve any ambiguities. SETUP,STORE, and PSW are modules used in the test program to set up ports and echo characters through the AN/UYK-20. 'TITLE' delimits a module. 'LOC' is an origin statement. 'EXTERNAL' denotes variables defined in other modules. 'GLOBAL' denotes variables to be used by other modules.

```
    1   00000  041000          NODEV  EQU     041000
    2   00000  040000          DATERR EQU     040000
    3   00000  100000          LUNUL  EQU     0100000        ;SIGNIFIES NO PORT IN LUMAP.
    4   00000  040400          FRNV   EQU     040400
    5   00000  001000          IDLE   EQU     01000
    6   00000  000000          R0     EQU     0              ;LITERALS FO  REGISTER REFERENCE.
    7   00000  000001          R1     EQU     1
    8   00000  000002          R2     EQU     2
    9   00000  000003          R3     EQU     3
   10   00000  000004          R4     EQU     4
   11   00000  000005          R5     EQU     5
   12   00000  000006          R6     EQU     6
   13   00000  000007          R7     EQU     7
   14   00000  000010          R8     EQU     8
   15   00000  000011          R9     EQU     9
   16   00000  000012          R10    EQU     10
   17   00000  000013          R11    EQU     11
   18   00000  000014          R12    EQU     12
   19   00000  000015          R13    EQU     13
   20   00000  000016          R14    EQU     14
   21   00000  000017          R15    EQU     15
   22   00000  000016          ENINT  EQU     016            ;ENABLE INTERRUPTS.
   23   00000  000244          DOL    EQU     0244           ;DOLLAR SIGN.
   24   00000  000000          CLEAR  EQU     0              ;USED IN CCR INSTRUCTION
   25   00000  000000          CH     EQU     0              ;MULTIPLEXOR CHANNEL-VARIES WITH SETUP.
   26   00000  000140          IOCELL EQU     0140
   27   00000  000010          MC     EQU     010            ;MASTER CLEAR
   28   00000  000040          LUMAX  EQU     32
   29   00000  000100          RESET  EQU     0100           ;RESET FOR USARTS
   30   00000  000001          OUTCHN EQU     01             ;PARAMETER FOR OCK
   31   00000  000000          INCHN  EQU     0              ;AND ICK INSTRUCTION    ;
   32   00000  177400          IOACTIV        EQU     0177400
   33   00000  177400          UPPER  EQU     0177400
   34   00000  000377          LOWER  EQU     0377
   35   00000  000015          MXFCN  EQU     015            ;FCN. REQUEST CODES.
   36   00000  000002          INFCN  EQU     02
   37   00000  000006          TMFCN  EQU     06
   38   00000  000011          STFCN  EQU     011
   39   00000  000012          CMDFCN EQU     012
   40                                                        ;NUMERIC LITERALS.
   41   00000  000000          ZERO   EQU     0
   42   00000  000002          TWO    EQU     2
   43   00000  000004          FOUR   EQU     4
   44   00000  000005          FIVE   EQU     5
   45                          INITSYMS       88
   46                          LOC    0110
   47                          FASTLOAD
   48                   PSW    TITLE
   49                          EXTERNAL       IOINT
   50   00000  000000                 +       0              ;STORE P LOC.
   51   00001  000000                 +       0              ;STORE SR1 LOC.
   52   00002  000000                 +       0              ;STORE SR2 LOC.
   53   00003  000000                 +       0              ;STORE RTC LOW LOC.
```



```
   54   00004  000000          INTAD  +       IOINT          ;INT. BASE ADDRESS.
```

```
   55   00005  000000                    +       0              ;LOC. FOR INT. SR1.
   56   00006  000000                    +       0              ;LOC. FOR INT. SR2.
   57   00007  000000                    +       0              ;STORE RTC UPPER LOC.
   58                                    END
1


                                MYASS VERSION 2.4/10 OF 12-18-75                PAGE   3

    1                           LOC     02000
    2                           FASTLOAD
    3                   STORE   TITLE
    4                           ;STORAGE AREAS AND VARIABLES.
    5                           EXTERNAL        PSW,RDINT,WRINT
    6                           GLOBAL  MD16,MD64,PKT,PKT1,PKT2,ZAP,ONCH0,IOINT
    7   00000  000316   MD16    +       0316           ;1 STOP,8 BITS,NO PAR., RESET ERROR FLG.
    8   00001  000067           +       067            ;SET DTR,RTS, ENABEL T/R,16X CK.
    9   00002  000117   MD64    +       0117           ;1 STOP,8 BITS,NO PAR., RESET ERROR FLG.
   10   00003  000067           +       067            ;SET DTR,RTS,ENABLE T/R,64X CK.
   11                           EVEN
   12                   PKT     RES     5
   13                           EVEN
   14                   PKT1    RES     5
   15                           EVEN
   16                   PKT2    RES     5
   17                           EVEN
   18   00026  70 0 00 10   ZAP     CCR     CH,MC
   19   00027  70 0 00 16   ONCH0   CCR     CH,ENINT       ;ENABLE CH 0 INTERRUPTS.
   20                           EVEN
   21   00030  40 2 10 00  000000   IOINT   JK      NOINT          ;IO INTERRUPT JUMP TABLE.
   22   00032  40 2 10 00  000000           JK      RDINT
   23   00034  40 2 10 00  000000           JK      WRINT
   24   00036  40 2 10 00  000000           JK      NOINT
   25                           RES     120            ;RESERVE SPACE FOR REST OF IO JUMP TABLE.
   26   00230  07 3 00 00  000000   NOINT   LP      PSW
   27                           END
1


                                MYASS VERSION 2.4/10 OF 12-18-75                PAGE   4

    1                           LOC     03000
    2                           FASTLOAD
    3                   SETUP   TITLE
    4                           EXTERNAL        ZAP,ONCH0,MD64,PKT,MD16,TEMP,PKT1,PKT2
    5                           EXTERNAL        MXS0,PSW,CHRDST,CHWRST
    6                                           ;SETS UP PORTS 2-4 THEN OUTPUTS
    7                                           ;'0' TO INDICATE SIGN ON, THEN ACTS IN ECHO
    8                                           ;MODE.
    9                                           ;
   10   00000  01 2 17 00  000016            LK      R13,ENINT      ;
   11   00002  03 0 17 05                    LSOR    R13            ;ENABLE INTERRUPTS.
   12   00003  01 3 06 00  000000            L       R6,ZAP
   13   00005  11 3 06 00  000140            S       R6,IOCELL      ;LOAD CLEAR CHANNELS TO IOCELL.
   14   00007  35 0 00 00                    IOCR
   15   00010  01 3 07 00  000140   WAIT     L       R7,IOCELL
   16   00012  47 2 07 00  000010            JNK     R7,WAIT        ;WAIT FOR IO INITIATION.
   17   00014  63 0 04 00                    LL      R4,ZERO
   18   00015  11 3 04 00  000000            S       R4,CHWRST      ;SET CH. STAT. INACTIVE.
   19   00017  11 3 04 00  000000            S       R4,CHRDST
   20   00021  01 3 06 00  000000            L        R6,ONCH0
   21   00023  11 3 06 00  000140            S       R6,IOCELL      ;LOAD CHANNEL 0 ON
   22   00025  35 0 00 03                    IOCR
   23   00026  01 3 07 00  000140   WAIT1    L       R7,IOCELL
```

```
24      00030  47 2 07 00  000026          JRK     R7,WAIT1
25                                                                ;CHANNEL 0 INTERRUPT ENABLED. OTHER CHANNELS
26                                                                ;DISABLED.
27                                                                ;
28                                                                ;FORM PACKET TO INITIALIZE PORT 2
29      00032  01 2 10 00  000002          LK      R8,TWO          ;INIT. FCN. REQUEST.
30      00034  01 2 11 00  000002          LK      R9,TWO          ;DEVICE TWO.
31      00036  01 2 12 00  000000          LK      R10,MD16        ;MD16 IS BUFFER ADDRESS.
32      00040  01 2 13 00  000002          LK      R11,TWO         ;BUFFER IS 2 WDS.
33      00042  01 3 14 00  000001          L       R12,MD16+1      ;LAST WD. READ FROM BUFFER.
34                                                                ;PACKET IS IN R8-R12.
35      00044  13 3 10 14  000000          SM      R8,PKT,R12      ;SAVE IN PACKET AREA.
36      00046  42 2 17 00  000000          JLRK    R15,MXSS        ;CALL IO HANDLER, LINK R15.
37      00050  000000                      +       PKT
38      00051  42 2 17 00  000000          JLRK    R15,TEST        ;TEST FOR IO DONE.
39      00053  000000                      +       PKT
40      00054  01 2 11 00  000004          LK      R9,FOUR         ;SAME PACKET FOR LU=4.
41      00056  13 3 10 14  000000          SM      R8,PKT1,R12     ;SAVE PACKET.
42      00060  42 2 17 00  000000          JLRK    R15,MXSS        ;CALL IO HANDLER.
43      00062  000000                      +       PKT1
44      00063  42 2 17 00  000000          JLRK    R15,TEST
45      00065  000000                      +       PKT1
46      00066  02 0 11 11                  DROR    R9              ;ADDRESS LU3.
47      00067  01 2 12 00  000000          LK      R10,MD16        ;LU3 WILL USE 16X CLOCK.
48      00071  01 3 14 00  000001          L       R12,MD16+1      ;FILL LAST WORD READ.
49      00073  13 3 10 14  000000          SM      R8,PKT2,R12     ;SAVE PACKET.
50      00075  42 2 17 00  000000          JLRK    R15,MXSS
51      00077  000000                      +       PKT2
52      00100  42 2 17 00  000000          JLRK    R15,TEST
53      00102  000000                      +       PKT2
```



```
54                                                                ;PORTS 2-4 ARE ACTIVE.
55                                                                ;OUTPUT SIGN ON SYMBOL '@'
56      00103  01 2 10 00  000001          LK      R8,01           ;WRITE FCN. REQUEST.
57      00105  01 2 11 00  000001          LK      R9,01           ;SET LU TO 1.
58      00107  01 2 12 00  000004          LK      R10,PKT+4       ;PACKET LAST WORD IS BUFFER.
59      00111  01 2 13 00  000001          LK      R11,01          ;ONE WORD BUFFER.
60      00113  01 2 14 00  000244          LK      R12,DOL         ;DOLLAR SIGN TO BE OUTPUT.
61      00115  02 0 11 10          LOOPS   IROR    R9              ;POINT TO NEXT LU.
62      00116  13 3 10 14  000000          SM      R8,PKT,R12
63      00120  42 2 17 00  000000          JLRK    R15,MXSS        ;CALL IO HANDLER.
64      00122  000000                      +       PKT
65      00123  42 2 17 00  000000          JLRK    R15,TEST
66      00125  000000                      +       PKT
67      00126  24 2 11 00  000004          CK      R9,FOUR
68      00130  40 2 03 00  000115          JLSK    LOOPS
69                                                                ;ON EXIT FROM LOOPS PORTS 2-4
70                                                                ;ARE SIGNED ON.
71      00132  01 2 12 00  000000          LK      R10,TEMP        ;TEMP IS BUFFER.
72      00134  01 2 13 00  000001          LK      R11,01          ;ONE WD. TRANSFER.
73      00136  13 3 10 14  000000          SM      R8,PKT,R12
74      00140  13 3 10 14  000000          SM      R8,PKT1,R12
75                                                                ;READ AND WRITE PACKETS ARE
76                                                                ;SETUP FOR IMAGE MODE.  NEXT
77                                                                ;LOOP SETS SEQUENTIAL ECHO.
78      00142  01 2 10 00  000013  ECHOS  LK      R8,013          ;REINITIALIZE PKT. FOR READ IM. REQ.
79      00144  11 3 10 00  000000          S       R8,PKT
80      00146  42 2 17 00  000000          JLRK    R15,MXSS        ;SETUP READ.
81      00150  000000                      +       PKT
82      00151  42 2 17 00  000000          JLRK    R15,TEST        ;WAIT ON IO DONE.
```

```
 83   00153  000000                        +       PKT
 84   00154  01 2 10 00  000014            LK      R8,014          ;REINITIALIZE STAT. FOR WR. IM. REQ.
 85   00156  11 3 10 60  000000            S       R8,PKT1
 86   00160  42 2 17 00  000000            JLRK    R15,MXSS        ;WRITE BACK TO PORT.
 87   00162  000000                        +       PKT1
 88   00163  42 2 17 00  000000            JLRK    R15,TEST        ;WAIT FOR IO.
 89   00165  000000                        +       PKT1
 90   00166  40 2 10 00  000142            JK      ECHOS
 91   00170  05 1 16 17              TEST    LXI     R14,R15         ;PKT. ADR. TO R14, SET RETURN ADR.
 92   00171  01 1 07 16              TEST1   LI      R7,R14          ;GET CONTENTS OF PKT. FST. WD.
 93   00172  30 2 07 00  177400            ANDK    R7,UPPER        ;MASK STATUS.
 94   00174  24 2 07 00  177400            CK      R7,IOACTIV      ;IF IO  ACTIVE,
 95   00176  40 2 00 00  000171            JEK     TEST1           ;RETEST PACKET STATUS.
 96   00200  40 0 10 17                    JR      R15
 97                                                               ;
 98                                                               ;INTERRUPT DOES NOT PERTAIN TO
 99                                                               ;MULTIPLEXOR CHANNEL.
100   00201  07 3 00 00  000000            LP      PSW
101                                        END
```



```
  1                                        LOC     04000
  2                                        FASTLOAD
  3                              MXSS      TITLE
  4                                        EXTERNAL        PSW
  5                                        GLOBAL  RDINT,WRINT,TEMP,CHRDST,CHWRST
  6                                                        ;IO DRIVER RO RS232 MULTIPLEXOR.
  7                                                        ;CALL: JLR R15, MXSS
  8                                                        ;R15 WILL CONTAIN POCKET ADDRESS
  9                                                        ;PACKET IS STANDARD FIVE WORD
 10                                                        ;PACKET THROUCH USAGE IS A BIT
 11                                                        ;DIFFERENT.  STATMP AND LUMAP MUST
 12                                                        ;BE SET UP PROPERLY WITH PORT STATUS AND
 13                                                        ;PHYSICAL ADDRESS, RESPECTIVELY.
 14                                                        ;MAX. NO. PORTS=32 FOR THIS VERSION MXSS
 15                                                        ;FUNCTION REQUESTS:  READ,READIM,
 16                                                        ;WRITE,WRITEIM,INIT,TERM,CMD,STAT.
 17                                                        ;ADDITIONAL FUNCTIONS ARE ADDED
 18                                                        ;VIA UNUSED FCN. MAP OPTIONS.
 19                                                        ;FUNCTION MAP OPTIONS.
 20   00000  13 3 00 16  000000             SM      R0,REGBNK,R14   ;SAVE REGISTERS EXCEPT LINK-R15
 21   00002  05 1 16 17                     LXI     R14,R15         ;PKT. ADR. TO R14, SET RETURN ADDRESS IN R15.
 22   00003  01 0 07 16                     LR      R7,R14          ;PKT.  ADR. TO R7.
 23   00004  06 1 10 07                     LDXI    R8,R7           ;GET PACKET CONTENTS.
 24   00005  06 1 12 07                     LDXI    R10,R7
 25   00006  01 1 14 07                     LI      R12,R7
 26   00007  30 2 11 00  000077            ANDK    R9, 077         ;MASK LU
 27   00011  01 3 07 11  000000            L       R7,LUMAP,R9     ;GET PHYS. ADR. AND STATUS.
 28   00013  30 2 10 00  000177            ANDK    R8,0177         ;MASK FCN. REQUEST.
 29   00015  24 2 10 00  000015            CK      R8,MXFCN        ;IF FCN. REQ. NOT VALID,
 30   00017  40 2 02 00  000000            JCEK    NVALID          ;RETURN INVALID STATUS.
 31   00021  46 3 07 10  000000            JP      R7,FCNMAP,R8    ;IF LU ACTIV., SWITCH TO FCN.
 32                                                        ;PORT IS IDLE OR NON-EXISTANT.
 33                                                        ;CHANNEL FCN.'S WILL BE SERVICED
 34                                                        ;IF THEY OCCUR.
 35   00023  24 2 10 00  000006            CK      R8,TMFCN
 36   00025  40 2 00 00  000000            JEK     TERM            ;IF TM. FCN. REQ., GO TO TERM.
 37   00027  24 2 10 00  000013            CK      R8,013          ;IF WRITIM OR READIM OCCUR,
 38   00031  40 3 02 10  000000            JCE     FCNMAP,R8       ;SWITCH TO REQUESTED FCN.
 39   00033  24 2 07 00  100000            CK      R7,LUNUL
 40   00035  40 2 00 00  000000            JEK     NONEX           ;IF PORT DOES NOT EXIST, GO TO NONEX.
```

```
41    00037  24 2 10 00  000002          CK      R8,INFCN        ;CHECK FOR INIT REQ.
42    00041  40 2 00 00  000000          JEK     INIT            ;IF INIT REQ.,GO TO INIT,
43    00043  31 2 10 00  001000          ORK     R8,IDLE         ;ELSE RETURN IDLE STATUS.
44    00045  11 1 10 16                  SI      R8,R14
45    00046  40 2 10 00  000000          JK      RETURN
46                                                             ;
47                                                             ;
48                                                             ;
49                                                             ;
50                                 WRITE                       ;WRITE REQUEST
51    00050  01 3 07 00  000000          L       R7,CHWRST
52    00052  47 2 07 00  000000          JNK     R7,BUSY         ;IF R7 NEG., CHANNEL IS BUSY.
53                                                             ;CHANNEL WRITE IS INACTIVE

                                MYASS VERSION 2.4/10 OF 12-18-75                    PAGE  7

54                                                             ;SETUP AND INITIATE WRITE
55                                                             ;
56    00054  01 0 07 13                  LR      R7, R11        ;
57    00055  02 0 07 11                  DROR    R7             ;(BUFFER WD. CNT.)-1 TO R7
58    00056  01 0 04 12                  LR      R4,R10         ;BUF. PTR. TO R4.
59    00057  01 2 03 00  177400          LK      R3,UPPER       ;SET MASK BITS FOR ADR. BYTE.
60    00061  01 1 02 04           LOOP   LI      R2,R4
61    00062  33 3 02 11  000000          MS      R2,LUMAP,R9    ;CONCATENATE PORT ADR.--DATA.
62    00064  15 1 02 04                  SXI     R2,R4          ;SAVE, INC. BUF. PTR.
63    00065  41 2 07 00  000061          XJK     R7,LOOP        ;
64                                                             ;
65                                                             ;BUFFER IS NOW FORMATTED IN
66                                                             ;PHYSICAL ADDRESS--DATA FORMAT
67                                                             ;
68                                                             ;SET UP CHANNEL FOR WRITE
69                                                             ;
70    00067  01 0 06 13           OUTPUT LR      R6,R11
71    00070  01 0 07 12                  LR      R7,R10
72    00071  05 0 06 17                  SBR     R6,R15         ;SETUP BCW FOR WORD TRANSFER, LOAD
73                                                             ;BUFFER COUNT AND POINTER
74    00072  12 3 06 00  000000          SD      R6,WRBCW       ;LOAD WRITE BUFFER CONTROL
75                                                             ;
76                                                             ;SET STATUS TO IO ACTIVE
77    00074  01 0 07 11                  LR      R7,R9
78    00075  31 2 07 00  177400          ORK     R7,IOACTIV
79    00077  11 3 07 00  000000          S       R7,CHWRST      ;SET CH. STATUS.
80    00101  11 3 16 00  000030          S       R14,SPKTWR     ;SAVE PKT.  ADR. IN PKT. WRITE STORE.
81    00103  31 2 10 00  177400          ORK     R8,IOACTIV     ;PKT. STAT. TO IOACTIV.
82    00105  11 1 10 16                  SI      R8,R14         ;IOACTIV TO PKT.
83                                                             ;
84    00106  02 3 06 00  000000          LD      R6,TXTCMD
85    00110  12 3 06 00  000140          SD      R6,IOCELL      ;LOAD IO WITH OUTPUT COMMAND
86    00112  35 0 00 00                  IOCR                   ;INITIATE IO
87                                                             ;
88                                                             ;
89                                 RETURN                      ;SET RETURN
90    00113  03 3 00 16  000000          LM      R0,RECBNK,R14
91    00115  40 0 10 17                  JR      R15
92    00116  70 0 00 16           TXTCHN CCR     CH,ENINT
93    00117  70 3 01 00  000000          IO      OUTCHN,WRBCW
94    00121  73 0 01 00                  IPR
95    00122  73 0 00 00                  HCR
96                                                             ;
97                                                             ;
98                                                             ;
99    00123  01 3 07 00  000000   WRITIM L      R7,CHWRST
```

```
100     00125   47 2 07 00  000000        JNK     R7,BUSY         ;TEST CHANNEL WRITE, EXIT IF BUSY
101                                                                 ;
102     00127   40 2 10 00  000067        JK      OUTPUT          ;OUTPUT IS SAME AS FOR WRITE
103                                                                 ;
104                                                                 ;
105                                                                 ;
106     00131   31 2 10 00  013000  BUSY  ORK     R8,013000       ; BUSY WORD TO STATUS
1

                                MYASS VERSION 2.4/10 OF 12-18-75                        PAGE   8

107     00133   11 1 10 16                SI      R8,R14          ;UPDATE PACKET.
108     00134   40 2 10 00  000113        JK      RETURN
109                                                                 ;
110                                                                 ;
111                                                                 ;
112                               READ                              ;READ OR READIM REQUEST.
113     00136   01 3 07 00  000000        L       R7,CHRDST
114     00140   47 2 07 00  000131        JNK     R7,BUSY         ;IF R7 NEG., CH. RD. IS BUSY    .
115                                                                 ;
116                                                                 ;SETUP CHANNEL READ
117                                                                 ;
118     00142   01 0 06 13        INPUT   LR      R6,R11
119     00143   01 0 07 12                LR      R7, R10
120     00144   05 0 06 17                SBR     R6,R15          ;SET BIT 15 - WORD TRANSFER
121     00145   12 3 06 00  000000        SD      R6,RDBCW        ;LOAD RDBCW.
122                                                                 ;SET STATUS=IOACTIV.
123     00147   31 2 11 00  177400        ORK     R9,IOACTIV
124     00151   11 3 11 00  000000        S       R9,CHRDST       ;SET THE NEXT READ STATUS
125     00153   11 3 16 00  000000        S       R14,SPKTRD      ;SAVE PACKET ADDRESS IN PKT. READ STORE.
126     00155   31 2 10 00  177400        ORK     R8,IOACTIV      ;SET PKT. STAT. TO IOACTIV.
127     00157   11 1 10 16                SI      R8,R14          ;STORE STATUS IN PKT. FST. WD.
128     00160   02 3 06 00  000000        LD      R6,RCVCMD
129     00162   12 3 06 00  000140        SD      R6,IOCELL       ;LOAD IOCELL.
130     00164   35 0 00 00                IOCR                    ;START IO CHAIN
131     00165   40 2 10 00  000113        JK      RETURN
132                                                                 ;
133                                                                 ;
134     00167   70 0 00 16        RCVCHN  CCR     CH,ENINT
135     00170   70 3 00 00  000000        IO      INCHN,RDBCW
136     00172   73 0 01 00                IPR
137     00173   73 0 00 00                HCR
138                                                                 ;
139                                                                 ;
140                                                                 ;
141                               CMD                               ;SEND COMMAND TO SPECIFIC PORT.
142     00174   01 3 07 00  000000        L       R7,CHWRST
143     00176   47 2 07 00  000131        JNK     R7,BUSY         ;EXIT IF CHANNEL BUSY
144     00200   01 0 07 13        CMDOK   LR      R7,R11
145     00201   02 0 07 11                DROR    R7              ;WORD COUNT-1 TO R7.
146     00202   01 0 04 12                LR      R4,R10          ;R4 SHALL BE BUFFER PTR.
147     00203   01 1 06 04        LOOP2   LI      R6,R4           ;GET DATA.
148     00204   30 2 06 00  000377        ANDK    R6,LOWER        ;MAKE SURE UPPER BYTE OF DATA WORD=0.
149     00206   63 0 02 00                LL      R2,ZERO         ;ZERO R2.
150     00207   24 2 06 00  000100        CK      R6,RESET        ;IS THERE A RESET COMMAND,
151     00211   40 2 01 00  000000        JNEK    NORESET         ;IF NOT, CONTINUE
152     00213   24 2 10 00  000002        CK      R8,INFCN        ;ELSE CHECK FOR INIT. REQ.
153     00215   40 2 00 00  000000        JEK     MDTST           ;IF INIT REQ. CHECK FOR MODE
154     00217   01 2 02 03  177400        LK      R2,IOACTIV      ;ELSE PREPARE TO ACTIVATE RESET FLAG.
155     00221   24 0 04 12        MDTST   CR      R4,R10          ;A RESET COMMAND ISSUED IN AN INIT
156     00222   40 2 01 00  000000        JNEK    RSTERR          ;REQUEST ABORTS INIT, RETURNS ERROR.
157     00224   11 3 02 00  000000        NORESET S       R2,RSTFLG       ;UPDATE RESET FLAG
158     00226   31 3 C6 11  000000        OR      R6,LUMAP,R9     ;CONCATENATE ADDRESS--DATA.
```

```
  159    00230  65 0 06 16                    SBR     R6,R14          ;FORM COMMAND ADDRESS.
1

                                    MYASS VERSION 2.4/10 OF 12-18-75                    PAGE  9

  160    00231  06 0 06 17                    ZBR     R6,R15          ;CLEAR IDLE BIT.
  161    00232  15 1 06 04                    SXI     R6,R4           ;SAVE IN BUFFER , INDEX BUFFER PTR.
  162    00233  41 2 07 00  000203            XJK     R7,LOOP2
  163                                                                ;WHOLE BUFFER IS FILLED WITH COMMANDS
  164                                                                ;
  165    00235  40 2 10 00  000067            JK      OUTPUT
  166                                                                ;
  167                                                                ;
  168                                                                ;
  169                                  STAT                          ;YIELDS SPECIFIC PORT STATUS.
  170    00237  01 3 07 11  000000            L       R7,LUMAP,R9     ;GET LU INFO.
  171    00241  46 2 07 00  000030            JPK     R7,ACTIV        ;IF LU POS. PORT IS ACTIVE.
  172    00243  31 2 10 00  001000            ORK     R8,IDLE         ;ASSUME  IDLE STATUS.
  173    00245  24 2 07 00  100000            CK      R7,LUNUL
  174    00247  40 2 01 00  000000            JNEK    SAVST           ;IF PORT EXISTS HERE, KEEP STATUS IDLE.
  175    00251  30 2 10 00  000377  NXST      ANDK    R8,LOWER        ;SET STAT. TO NODEVICE.
  176    00253  31 2 10 00  041000            ORK     R8,NODEV        ;SET NO DEVICE STATUS
  177    00255  11 1 10 16          SAVST     SI      R8,R14          ;SAV PKT STATUS.
  178    00256  40 2 10 00  000113            JK      RETURN
  179    00260  01 3 14 11  000000  ACTIV     L       R12,MCMAP,R9    ;GET MODE--CMD. STATUS.
  180    00262  31 2 10 00  177400            ORK     R8,IOACTIV      ;ASSUME IO ACTIVE.
  181    00264  01 3 06 00  000000            L       R6,CHWRST
  182    00266  30 2 06 00  000377            ANDK    R6,LOWER        ;MASK FCN.
  183    00270  24 0 06 11                    CR      R6,R9
  184    00271  40 2 00 00  000000            JEK     IOON            ;IF CH. WRITE ON THIS PORT,
  185                                                                ;SET STATUS TO IOACTIV.
  186    00273  01 3 06 00  000000            L       R6,CHRDST
  187    00275  30 2 06 00  000377            ANDK    R6,LOWER        ;MASK FCN.
  188    00277  24 0 06 11                    CR      R6,R9
  189    00300  40 2 00 00  000000            JEK     IOON            ;IF CHANNEL READ ON THIS PORT,
  190                                                                ;SET STATUS TO IOACTIV.
  191    00302  30 2 10 00  000377  IOOFF     ANDK    R8,LOWER
  192    00304  01 0 04 16          IOON      LR      R4,R14          ;PKT. ADR. TO R4.
  193    00305  16 1 10 04                    SDXI    R8,R4           ;UPDATE PACKET.
  194    00306  02 0 04 12                    IRTR    R4
  195    00307  11 1 14 04                    SI      R12,R4          ;WRITE MD.--CMD. TO PACKET LAST WD.
  196    00310  40 2 10 00  000113            JK      RETURN
  197                                                                ;
  198                                                                ;
  199                                                                ;
  200                                  TERM                          ;TERMINATES ALL EXTANT CHANNELS
  201                                                                ;RESETS ALL PORTS IDLE
  202    00312  63 0 04 00                    II      R4,ZERO         ;ZERO R4.
  203    00313  01 3 07 00  000000            L       R7,ZAP
  204    00315  11 3 07 00  000140            S       R7,IOCELL
  205    00317  35 0 00 00                    IOCR                    ;RESET CHANNEL
  206    00320  01 2 07 00  000037            LK      R7,LUMAX-1      ;INDEX-1 TO R7.
  207    00322  01 3 06 07  000030  LOOP1     L       R6,LUMAP,R7     ;GET LUMAP.
  208    00324  24 2 06 00  100000            CK      R6,LUNUL
  209    00326  40 2 00 00  000000            JEK     NEXT            ;IF NO DEVICE-LOOK AT NEXT LU.
  210                                  EXIST                         ;IF PORT EXISTS , RESET IT.
  211    00330  06 0 06 17                    ZBR     R6,R15          ;TURN OFF IDLE BIT.
  212    00331  65 0 06 16                    SBR     R6,R14          ;FORM COMMAND ADDRESS.
1

                                    MYASS VERSION 2.4/10 OF 12-18-75                    PAGE  10
```

```
213     00332  31 2 06 00  000100          ORK     R6,RESET        ;CONCATENATE ADDRESS AND RESET COMMAND
214     00334  11 3 06 04  000000          S       R6,TBUF,R4      ;STORE IN TBUF INDEXED BY R4
215     00336  02 0 04 10                   IROR    R4              ;COUNT  OUTPUT COMMANDS.
216     00337  41 2 07 00  000322  NEXT    XJK     R7,LOOP1        ;LOOK AT NEXT PORT.
217                                                                 ;COUNT OF COMMAND IS IN R4
218                                                                 ;
219                                                                 ;
220     00341  05 0 04 17                   SBR     R4,R15          ;FORM BCW--WORD TRANSFER.
221     00342  01 2 05 00  000000          LK      R5,TBUF
222     00344  12 3 04 00  000000          SD      R4,WRBCW
223                                                                 ;OUTPUT
224     00346  40 2 10 00  000067          JK      OUTPUT
225                                                                 ;
226                                                                 ;
227                                                                 ;
228                                INIT                             ;RESET PORT. OUTPUT MODE FOLLOWED
229                                                                 ;BY SEQUENCE OF COMMANDS.
230     00350  01 3 07 00  000000          L       R7,CHWRST
231     00352  47 2 07 00  000131          JNK     R7,BUSY         ;EXIT IS CHANNEL WRITE BUSY.
232     00354  01 3 07 11  000000          L       R7,LUMAP,R9
233     00356  05 0 07 16                   SBR     R7,R14          ;SET BIT PORTION FOR COMMAND
234     00357  06 0 07 17                   ZBR     R7,R15          ;CLEAR IDLE BIT.
235     00360  31 2 07 00  000100          ORK     R7,RESET        ;FORM RESET COMMAND.
236     00362  11 3 07 00  000000          S       R7,RBUF         ;STORE IN BUFFER
237     00364  02 3 06 00  000000          LD      R6,RSTCMD
238     00366  12 3 06 00  000140          SD      R6,IOCELL       ;LOAD RST COMMAND
239     00370  63 0 07 00                   LL      R7,ZERO         ;ZERO R7.
240     00371  11 3 07 00  000000          S       R7,TEMP         ;LOAD TEMP WITH ZERO.
241     00373  35 0 00 00                   IOCR                    ;INITIATE IO
242     00374  01 3 07 00  000000  HOLD    L       R7,TEMP
243     00376  46 2 07 00  000374          JPK     R7,HOLD         ;HOLD FOR COMPLETION OF RESET
244                                                                 ;
245                                                                 ;CHANNEL IS NOT BUSY NOW.
246     00400  40 2 10 00  000200          JK      CMDOK           ;OUTPUT MODE AND COMMANDS CMDOK WILL
247                                                                 ;UPDATE COMMAND
248                                                                 ;
249                                                                 ;
250     00402  70 3 01 00  000000  RSTCHN  IO      OUTCHN,RSTBCW
251     00404  73 3 01 00  000000          SF      TEMP            ;IO COMPLETE-MAKE TEMP NEGATIVE.
252     00406  73 0 00 00                   HCR
253                                                                 ;
254                                                                 ;
255                                                                 ;
256                                NONEX                            ;NO PORT EXISTS WITH THIS LU.
257     00407  31 2 10 00  041000          ORK     R8,NODEV        ;OR IN NODEV STAT.
258     00411  11 1 10 16                   SI      R8,R14          ;STORE STATUS IN PACKET FOR WD
259     00412  40 2 10 00  000113          JK      RETURN          ;RETURN
260
261                                NVALID                           ;INVALID FCN. REQUEST
262     00414  31 2 10 00  040400          ORK     R8,FRNV         ;OR IN FUNCTION REQUEST NOT VALID
263     00416  11 1 10 16                   SI      R8,R14          ;STORE STAT IN PACKET FST. WD.
264     00417  40 2 10 00  000113          JK      RETURN          ;RETURN
265                                                                 ;
```



```
266                                RSTERR                           ;RESET WAS ISSUED IN INIT FCN. REQ.
267                                                                 ;ABORT INIT, RETURN STATUS TO PACKET.
268     00421  31 2 10 00  040000          ORK     R8,DATERR
269     00423  11 1 10 16                   SI      R8,R14          ;UPDATE PACKET WITH DATA ERROR STATUS.
270     00424  40 2 10 00  000113          JK      RETURN
271                                                                 ;
```

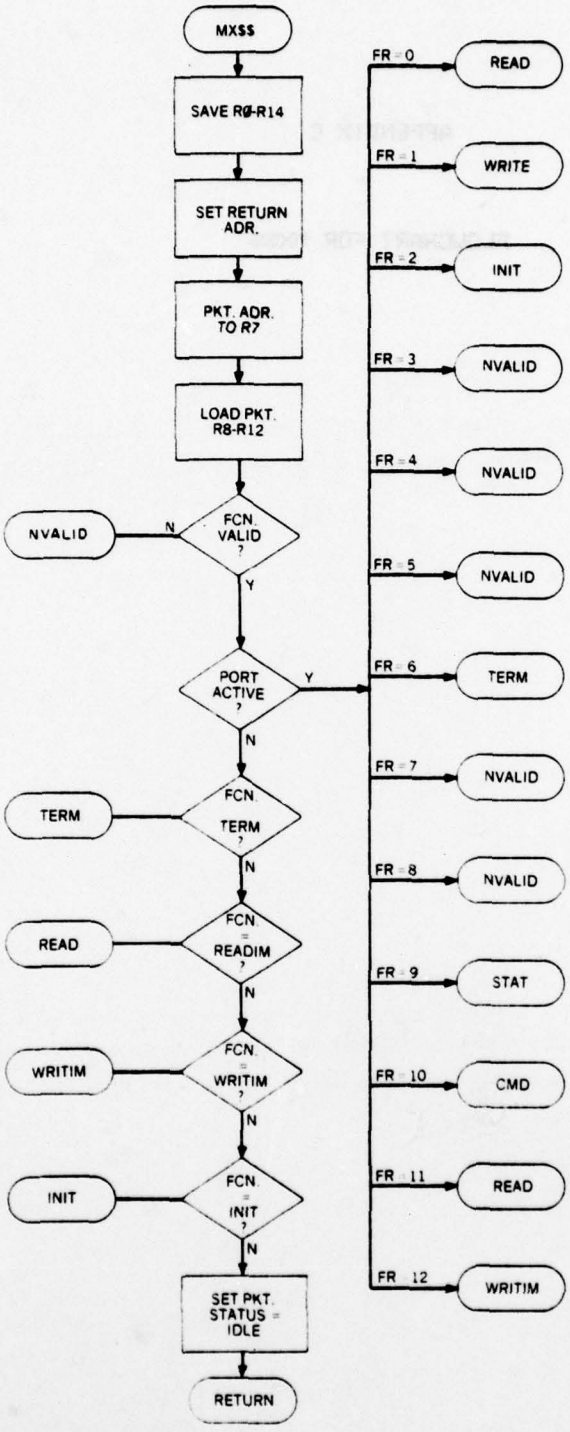
```
272                                                         ;
273                                 WRINT                   ;WRITE INTERRUPT. SET ALL STATUS TO
274                                                         ;FUNCTION COMPLETE.
275                                                         ;UPDATE MCMAP FOR CMD, INIT.
276                                                         ;SETS PORTS IDLE FOR TERM.
277  00426  13 3 00 17  000000          SM      R0,INTBNK,R15
278  00430  01 3 07 00  000000          L       R7,CHWRST       ;LOAD CH. STAT.
279  00432  01 2 06 00  177400          LK      R6,UPPER        ;UPPER BYTE MASK TO R6.
280  00434  30 0 06 07                  ANDR    R6,R7           ;CH. STAT. TO R6.
281  00435  30 2 07 00  000377          ANDK    R7,LOWER        ;PORT ADDRESS IS IN R7.
282  00437  01 3 16 00  000000          L       R14,SPKTWR      ;GET PACKET ADDRESS.
283  00441  01 0 04 16                  LR      R4,R14          ;PKT.ADR.  TO R4.
284  00442  06 1 10 04                  LDXI    R8,R4           ;GET PACKET CONTENTS.
285  00443  06 1 12 04                  LDXI    R10,R4
286  00444  01 1 14 04                  LI      R12,R4
287  00445  30 2 10 00  000377          ANDK    R8,LOWER        ;MASK PKT.FCN. REQ.
288  00447  40 3 10 10  000000          J       STATMP,R8       ;SWITCH TO APPROPRIATE STATUS
289  00451  000000      STATMP  +       NOSTAT
290  00452  000000              +       NOSTAT
291  00453  000000              +       INSTAT
292  00454  000000              +       NOSTAT
293  00455  000000              +       NOSTAT
294  00456  000000              +       NOSTAT
295  00457  000000              +       TSTAT
296  00460  000000              +       NOSTAT
297  00461  000000              +       NOSTAT
298  00462  000000              +       NOSTAT
299  00463  000000              +       CMDSTAT
300  00464  000000              +       NOSTAT
301  00465  000000              +       NOSTAT
302  00466  30 2 10 00  000377  NOSTAT  ANDK    R8,LOWER        ;REQ. COMPLETE STAT TO R8.
303  00470  11 1 10 16                  SI      R8,R14          ;UPDATE PACKET
304  00471  63 0 10 00                  LL      R8,ZERO         ;ZERO R8
305  00472  11 3 10 00  000000          S       R8,CHWRST       ;SET WRITE BUF. 0.(INACTIVE).
306  00474  03 3 00 17  000000          LM      R0,INTBNK,R15   ;STORE REGISTERS
307  00476  07 3 00 00  000000          LP      PSW             ;RETURN FROM INTERRUPT
308
309                                                         ;
310                                 TSTAT                   ;UPDATE STATUS FOR TERM COMMAND.
311  00500  01 2 07 00  000037          LK      R7,LUMAX-1
312  00502  63 0 06 00                  LL      R6,ZERO
313  00503  01 3 10 07  000000  IDLOOP  L       R8,LUMAP,R7
314  00505  05 0 10 17                  SBR     R8,R15
315  00506  11 3 10 07  000000          S       R8,LUMAP,R7     ;STORE IDLED LU.
316  00510  11 3 06 07  000000          S       R6,MCMAP,R7     ;ZERO MCMAP.
317  00512  41 2 07 00  000503          XJK     R7,IDLOOP       ;IDLE ALL PORTS.
318                                                         ;
1

                                    MYASS VERSION 2.4/10 OF 12-18-75                    PAGE  12

319  00514  40 2 10 00  000466          JK      NOSTAT
320                                                         ;
321                                 INSTAT                  ;UPDATE INIT STATUS.
322  00516  01 1 04 12                  LI      R4,R10          ;GET MODE(FST. BUF. WD.)
323  00517  01 0 05 14                  LR      R5,R12          ;GET CMD.(PKT. LST. WD. RD.)
324  00520  61 0 04 10                  LALS    R4,R8           ;MODE TO UPPER BYTE.
325  00521  31 0 04 05                  ORR     R4,R5           ;CONCATENATE MODE--CMD.
326  00522  11 3 04 07  000000          S       R4,MCMAP,R7     ;UPDATE MCMAP .
327  00524  01 3 04 07  000000          L       R4,LUMAP,R7     ;DE-IDLE PORT.
328  00526  06 0 04 17                  ZBR     R4,R15          ;RESET IDLE BIT.
329  00527  11 3 04 07  000000          S       R4,LUMAP,R7     ;SAVE DE-IDLED ADDRESS.
330                                                         ;
```

```
331     00531  40 2 10 00  000466          JK      NOSTAT
332                                                 ;
333                                 CMDSTAT                 ;UPDATE CMD. STATUS.
334     00533  01 3 04 07  000000          L       R4,MCMAP,R7     ;GET CURRENT MC.
335     00535  30 2 04 00  177400          ANDK    R4,UPPER        ;SAVE MODE
336     00537  31 0 04 14                  ORR     R4,R12          ;CONCATENATE MODE--LAST CMD.(@PKT + 4.)
337     00540  11 3 04 07  000000          S       R4,MCMAP,R7     ;UPDATE MCMAP.
338     00542  01 3 02 00  000000          L       R2,RSTFLG
339     00544  44 2 02 00  000466          JZK     R2,NOSTAT       ;IF RESET FLAG NOT SET EXIT,
340     00546  01 3 04 07  000000          L       R4,LUMAP,R7
341     00550  05 0 04 17                  SBR     R4,R15          ;ELSE UPDATE PORT STATUS TO IDLE.
342     00551  11 3 04 07  000000          S       R4,LUMAP,R7
343     00553  63 0 02 00                  LL      R2,ZERO         ;
344     00554  11 3 02 00  000000          S       R2,RSTFLG       ;RESET FLAG TO 0.
345     00556  40 2 10 00  000466          JK      NOSTAT
346                                 RDINT                   ;READ INTERRUPT--SETS PORT TO FCN.
347                                                         ;COMPLETE.  UPDATES PACKET.
348     00560  13 3 00 17  000000          SM      R0,INTBNK,R15   ;SAVE REGISTERS.
349     00562  01 3 16 00  000000          L       R14,SPKTRD      ;LOAD PKT.ADR. TO R14.
350     00564  01 1 10 16                  LI      R8,R14          ;PKT. FST. WD. TO R8.
351     00565  30 2 10 00  000377          ANDK    R8,LOWER        ;FORM FCN. COMPLETE STAT.
352     00567  11 1 10 16                  SI      R8,R14          ;UPDATE PKT.
353     00570  63 0 10 00                  LL      R8,ZERO         ;ZERO R8
354     00571  11 3 10 00  000000          S       R8,CHRDST       ;MAKE READ STAT. 0.(INACTIVE).
355     00573  03 3 00 17  000000          LM      R0,INTBNK,R15,
356     00575  07 3 00 00  000000          LP      PSW             ;RETURN FROM INTERRUPT
357
358                                                         ;
359                                                         ;
360
361                                                         ;
362                                                         ;
363                                                         ;
364                                                         ;
365                                 LUMAP                   ;MUST SET LUMAP FOR EACH CONFIGURATION
366                                                         ; OF MULTIPLEXOR.
367                                                         ;PHYSICAL PORT ADDRESS CONTAINED IN BITS 0-14.
368                                                         ;BIT 15 SET MEANS PORT IS IDLE.
369                                                         ;INITIALIZE WITH ALL PORTS IDLE.
370                                                         ;ADDRESS 0 IS INVALID PHYSICAL PORT ADR.
371                                                         ;AND SIGNIFIES PORT DOES NOT EXIST.
1

                                    MYASS VERSION 2.4/10 OF 12-18-75                    PAGE  13

372     00577  110000                      +       0110000
373     00600  110400                      +       0110400
374     00601  011000                      +       0011000
375     00602  111400                      +       0111400
376     00603  112000                      +       0112000
377     00604  112400                      +       0112400
378     00605  100000                      +       0100000         ;PORT DOES NOT EXIST.
379     00606  100000                      +       0100000
380     00607  100000                      +       0100000
381     00610  100000                      +       0100000
382     00611  100000                      +       0100000
383     00612  100000                      +       0100000
384     00613  100000                      +       0100000
385     00614  100000                      +       0100000         ;PORT DOES NOT EXIST.
386     00615  100000                      +       0100000
387     00616  100000                      +       0100000
388     00617  100000                      +       0100000
389     00620  100000                      +       0100000
```

```
390     00621  100000                    +       0100000
391     00622  100000                    +       0100000
392     00623  100000                    +       0100000
393     00624  100000                    +       0100000
394     00625  100000                    +       0100000
395     00626  100000                    +       0100000         ;PORT DOES NOT EXIST.
396     00627  100000                    +       0100000
397     00630  100000                    +       0100000
398     00631  100000                    +       0100000
399     00632  100000                    +       0100000
400     00633  100000                    +       0100000
401     00634  100000                    +       0100000
402     00635  100000                    +       0100000
403     00636  100000                    +       0100000
404     00637  000000            CHRDST  +       000000          ;INITIALIZE CH. BUFFERS TO INACTIVE.
405     00640  000000            CHWRST  +       000000
406                                      EVEN
407     00642  000000            SPKTWR  +       0
408     00643  000000            SPKTRD   +      0
409     00644  71 2 00 06  000116 TXTCMD OCK     CH,TXTCHN
410     00646  71 2 00 06  000402 RSTCMD OCK     CH,RSTCHN
411     00650  71 2 00 02  000167 RCVCMD ICK     CH,RCVCHN
412                              TEMP    RES     1
413                              RSTFLG  RES     1
414     00654  100001    RSTBCW  +       0100001
415     00655  000000            +       RBUF
416                              RECBNK  RES     16              ;REGISTER SAVE AREA.
417                              INTBNK  RES     16              ;INTERRUPT REGISTER SAVE AREA.
418     00716  000136            FCNMAP  +       READ
419     00717  000050                    +       WRITE
420     00720  000350                    +       INIT
421     00721  000414                    +       NVALID
422     00722  000414                    +       NVALID
423     00723  000414                    +       NVALID
424     00724  000312                    +       TERM
1

                                MYASS VERSION 2.4/10 OF 12-18-75                PAGE  14

425     00725  000414                    +       NVALID
426     00726  000414                    +       NVALID
427     00727  000237                    +       STAT
428     00730  000174                    +       CMD
429     00731  000136                    +       READ
430     00732  000123                    +        WRITIN
431                                      EVEN
432                              MCMAP   RES     32
433     00774  000000            RDBCW   +        0
434     00775  000000                    +       0
435     00776  000000            WRBCW   +       0
436     00777  000000                    +       0
437                                      EVEN
438     01000  70 0 00 10          ZAP     CCR     CH,MC
439                                      EVEN
440                              TBUF    RES     32
441                              RBUF    RES     1
442                                      END
1

                                MYASS VERSION 2.4/10 OF 12-18-75                PAGE  15
```

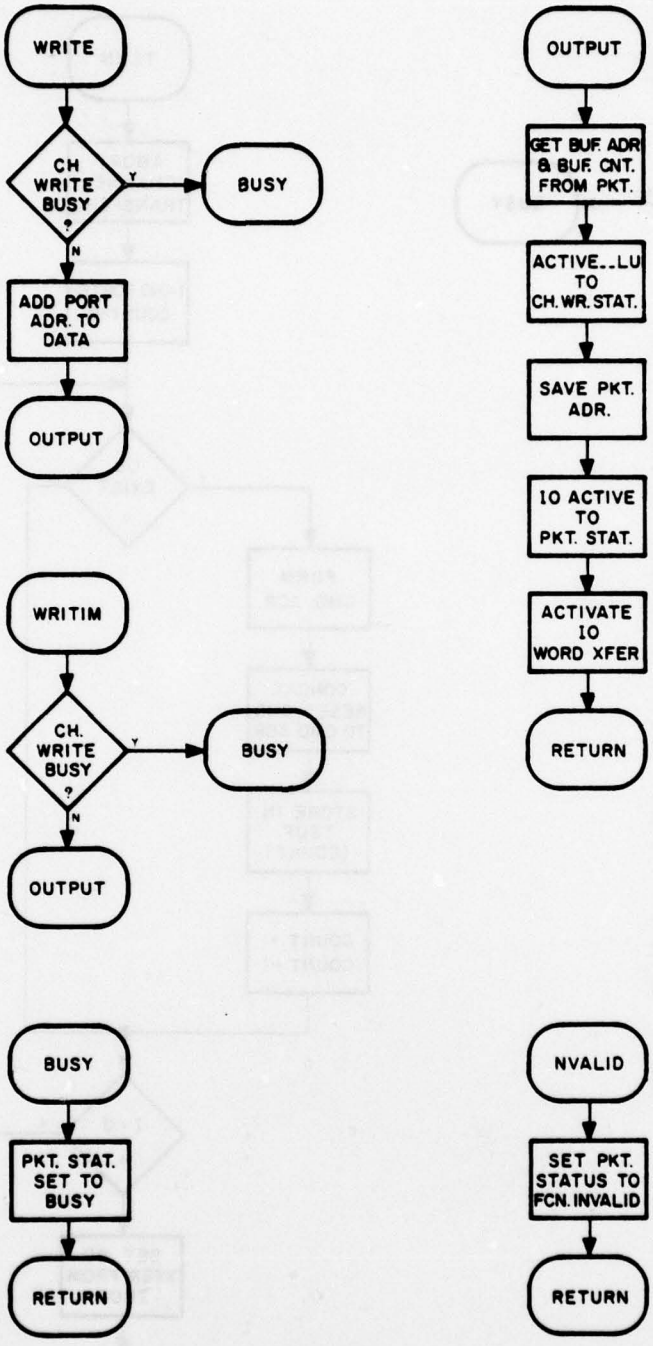
APPENDIX C

FLOWCHART FOR MXSS

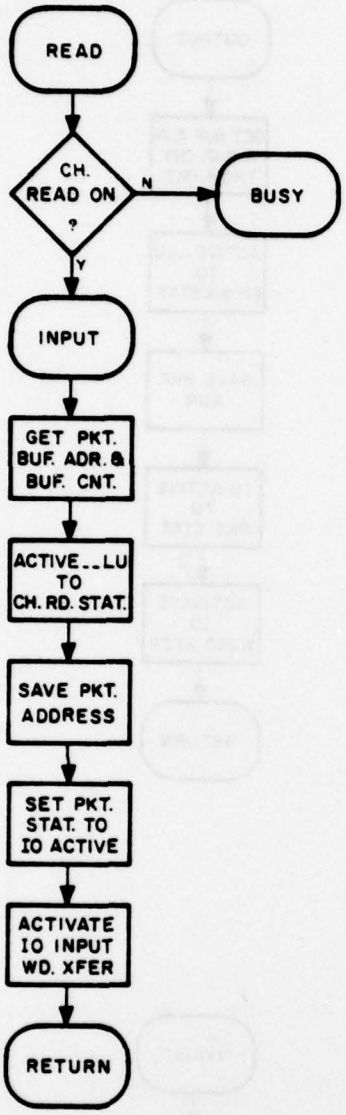


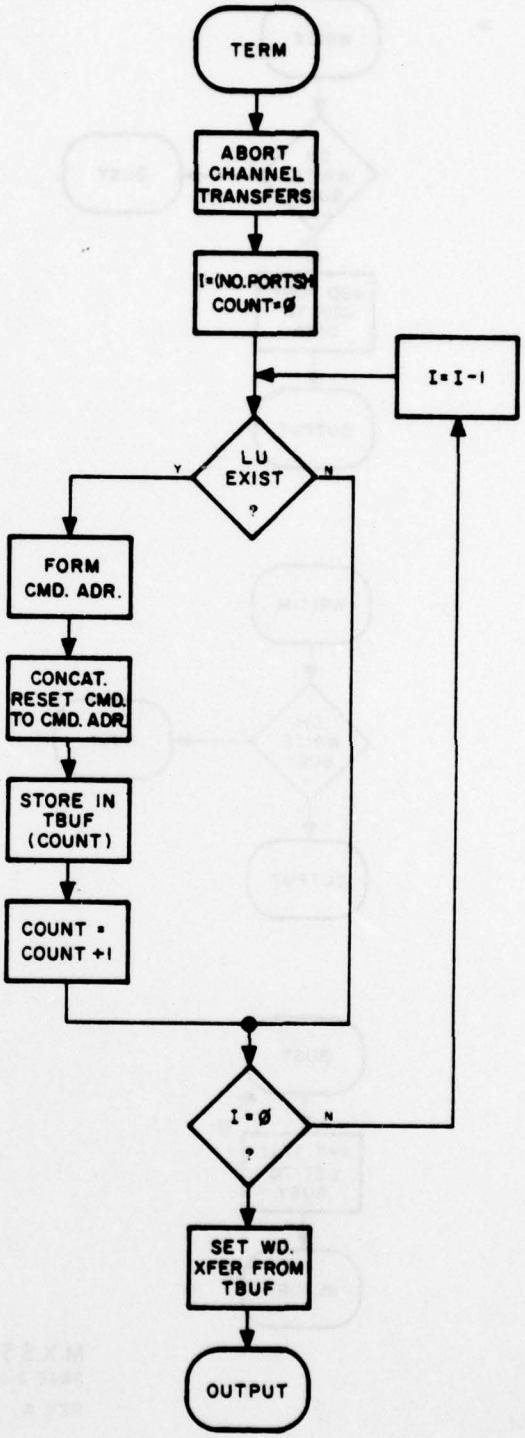
MX$$
PAGE 1 of 6

REV A 1.3.77
REV B 1.5.77



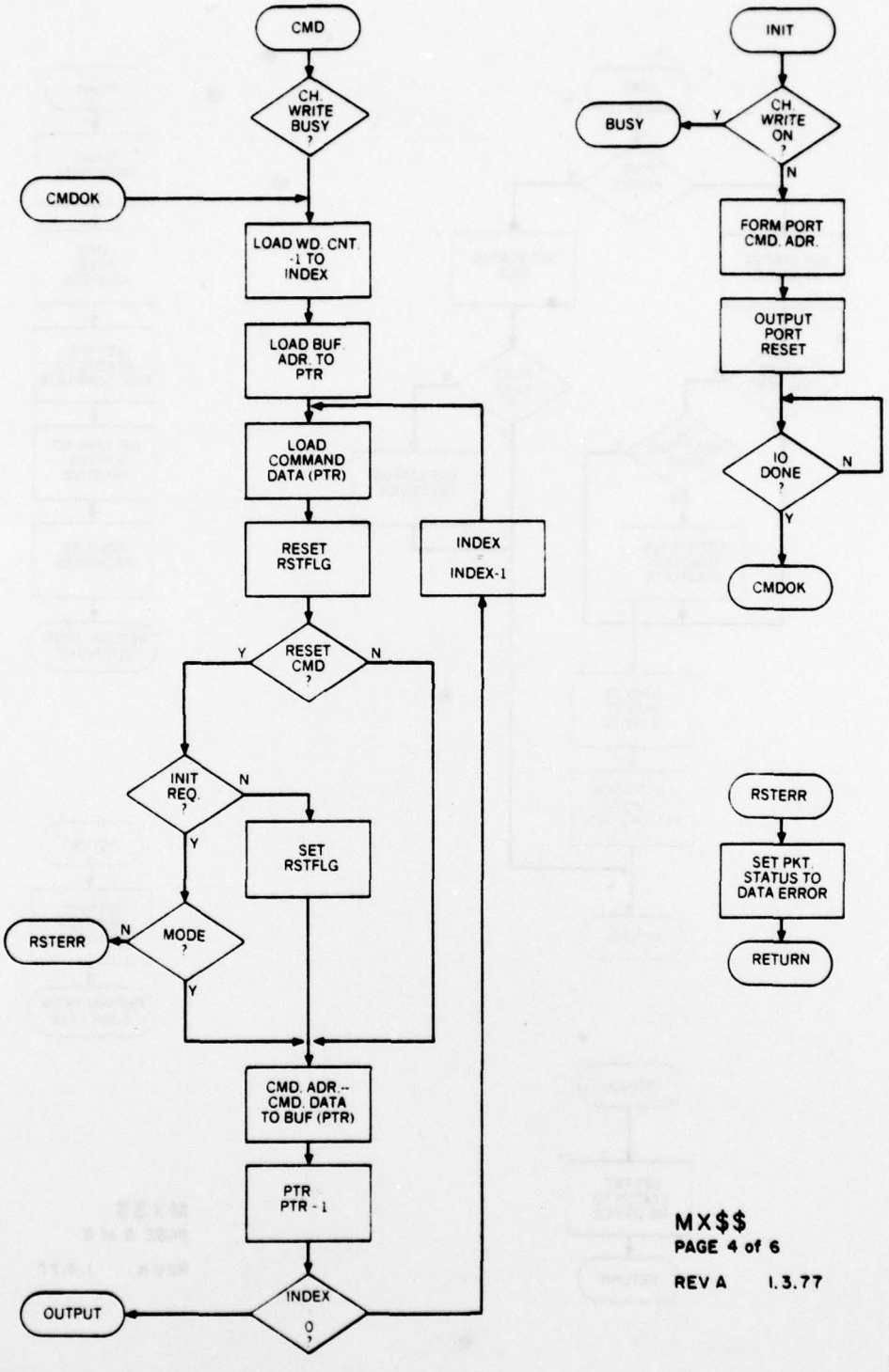
MX$$
PAGE 2 of 6
REV. A 1.3.77



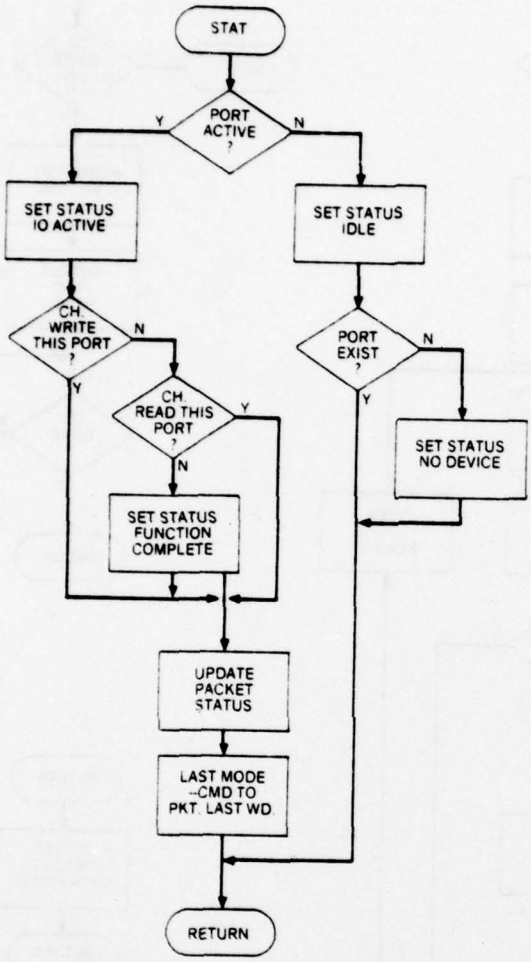


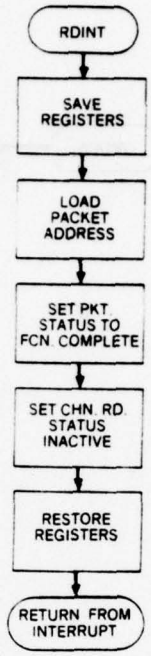
MX$$
PAGE 3 of 6

REV. A 1.3.77

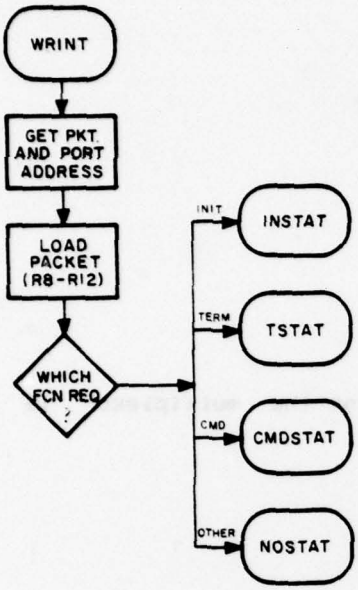


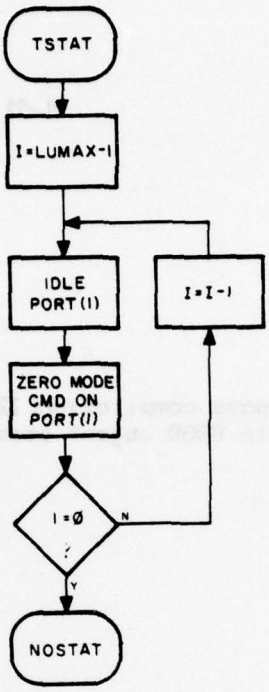
MX$$
PAGE 4 of 6
REV A 1.3.77

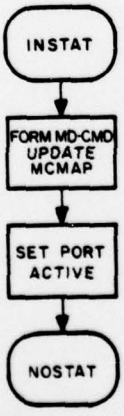


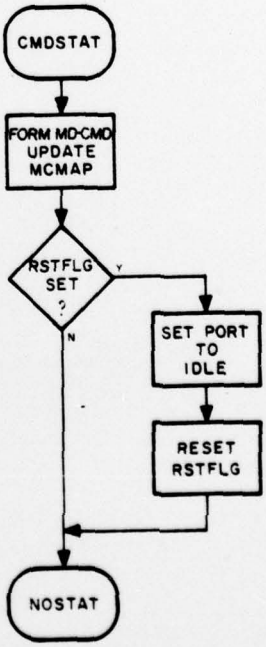


MX$$
PAGE 5 of 6
REV A 1.4.77









MX$$
PAGE 6 of 6
REV. A 1.4.77

APPENDIX D

PL/M DRIVER FOR MULTIPLEXOR

The two-pass compilation[12] of the PL/M driver for the multiplexor is shown with 8080 object code output[13].

```
16:02:05 BAJOB  BATCON VERSION 13(1071) RUNNING TXTOUT SEQUENCE 3304 IN STREAM 1
16:02:05 BAFIL  INPUT FROM DSKC0:TXT.CTL[253,533]
16:02:05 BAFIL  OUTPUT TO  DSKC0:TXTOUT.LOG[253,533]
16:02:05 BASUM  JOB PARAMETERS
                TIME:00:05:00   UNIQUE:YES       RESTART:NO

16:02:05 MONTR
16:02:05 MONTR  .LOGIN 253/533 /SPOOL:ALL/TIME:300/NAME:"LRUSSO"
16:02:08 USER   JOB 24 NRL-602-10 TTY65
16:02:08 USER   [LCNJSP OTHER JOBS SAME PPN:27]
16:02:08 USER   1602     26-JAN-77        WED
16:02:17 USER   NO MAIL
16:02:17 MONTR
16:02:17 MONTR  .COPY FOR20.DAT=*.TXT
16:02:28 MONTR
16:02:28 MONTR  .R PLMB1
16:02:28 USER
16:02:32 USER   8080 PLM1 VERS 3.0
16:02:36 USER
16:02:36 USER
16:02:36 USER   $I=6 $$
16:02:39 USER   $A=0
16:02:39 USER   $B=1
16:02:39 USER   $C=0
16:02:39 USER   $D=120
16:02:39 USER   $E=0
16:02:39 USER   $G=0
16:02:39 USER   $I=6
16:02:39 USER   $J=6
16:02:39 USER   $K=72
16:02:39 USER   $L=1
16:02:39 USER   $M=1
16:02:39 USER   $O=1
16:02:39 USER   $P=1
16:02:39 USER   $R=72
16:02:39 USER   $S=0
16:02:39 USER   $T=1
16:02:39 USER   $U=7
16:02:39 USER   $V=72
16:02:39 USER   $W=72
16:02:39 USER   $Y=1
16:02:39 USER
16:02:46 USER   00001  1   /*INTLAT.TXT 1.12.77            */
16:02:46 USER
16:02:46 USER   00002  1   /*THE FOLLOWING IS A BASIC PROGRAM FOR COMMUNICATION
16:02:46 USER
16:02:46 USER   00003  1    WITH THE AN-UYK20 DRIVER MX$$. AS AN EXAMPLE,
16:02:46 USER
16:02:46 USER   00004  1    PORT 2 IS ACTIVATED LOCALLY; PORTS 3,4 ARE PUT IN CO
M-
16:02:46 USER
16:02:46 USER   00005  1    MAND MODE. A SIGN-ON MESSAGE IS OUTPUT LOCALLY TO
16:02:46 USER
16:02:46 USER   00006  1    PORT2.*/
16:02:46 USER
16:02:46 USER   00007  1   DECLARE LIT LITERALLY 'LITERALLY';
16:02:46 USER
16:02:46 USER   00008  1   DECLARE DCL LIT 'DECLARE';
16:02:46 USER
16:02:53 USER   00009  1   DCL URT2 LIT '8012H';
16:02:53 USER
16:02:53 USER   00010  1   DCL URT3 LIT '8013H';
16:02:53 USER
```

```
16:02:53 USER    00011  1   DCL URT4 LIT '8014H';
16:02:53 USER
16:02:53 USER    00012  1   DCL MODE1 LIT '4EH';
16:02:53 USER
16:02:53 USER    00013  1   DCL ONLINE LIT '37H'; /*SET RTS,DTR,ENABLE T/R,RESET
ERRORS*/
16:02:53 USER
16:02:53 USER    00014  1   DCL MODE2 LIT '4FH';
16:02:53 USER
16:02:53 USER    00015  1   DCL MEMMAP LIT'8000H';
16:03:00 USER
16:03:00 USER    00016  1   DCL LOW7 LIT '7FH';
16:03:00 USER
16:03:00 USER    00017  1   DCL PARITY$IS$EVEN$ LIT 'PARITY';
16:03:00 USER
16:03:00 USER    00018  1   DCL MODE3 LIT '41H'; /*ASYNCH,5 BIT.,NO PARITY,1X*/
16:03:00 USER
16:03:00 USER    00019  1   DCL ERESET LIT '10H'; /*USRT ERROR FLAG RESET*/
16:03:00 USER
16:03:00 USER    00020  1   DCL WAIT$FOR$INTERRUPT LIT
16:03:00 USER
16:03:00 USER    00021  1    ' ENABLE;
16:03:00 USER
16:03:00 USER    00022  1    WAIT: GO TO WAIT;';
16:03:00 USER
16:03:00 USER    00023  1   DCL READY LIT '(CMD AND 01) > 0';
16:03:00 USER
16:03:11 USER    00024  1   DCL CMDBIT LIT '40H';
16:03:11 USER
16:03:11 USER    00025  1   DCL UYKAD ADDRESS;
16:03:11 USER
16:03:11 USER    00026  1   DCL (UYK20 BASED UYKAD) (2) BYTE;
16:03:11 USER
16:03:11 USER    00027  1   DCL CMDAD ADDRESS;
16:03:11 USER
16:03:11 USER    00028  1   DCL (CMD BASED CMDAD) BYTE;
16:03:11 USER
16:03:11 USER    00029  1   DCL IOINTADR ADDRESS;
16:03:11 USER
16:03:11 USER    00030  1   DCL (IOINT BASED IOINTADR) BYTE;
16:03:11 USER
16:03:11 USER    00031  1   DCL PORTPTR ADDRESS;
16:03:11 USER
16:03:11 USER    00032  1   DCL (PORT BASED PORTPTR) BYTE;
16:03:11 USER
16:03:20 USER    00033  1   DCL TEMP (2) BYTE;  /*TEMPORARY STORAGE*/
16:03:20 USER
16:03:20 USER    00034  1   DCL MESSAGE DATA ('$PORT2$',0DH,0AH);
16:03:20 USER
16:03:20 USER    00035  1   SETUP: PROCEDURE (PORTPTR,MDCMD,CDCMD);
16:03:20 USER
16:03:20 USER    00036  2   /*SENDS A MODE AND COMMAND INSTRUCTION TO APPROPRIATE
 PORT*/
16:03:20 USER
16:03:20 USER    00037  2   DCL PORTPTR ADDRESS;
16:03:20 USER
16:03:20 USER    00038  2   DCL (PORT BASED PORTPTR) BYTE;
16:03:20 USER
16:03:20 USER    00039  2   DCL (MDCMD,CDCMD) BYTE;
16:03:20 USER
16:03:20 USER    00040  2    CMDAD=PORTPTR OR CMDBIT;
16:03:28 USER
16:03:28 USER    00041  2    CMD=MDCMD;
16:03:28 USER
```

```
16:03:28 USER   00042  2    CMD=CDCMD;
16:03:28 USER
16:03:28 USER   00043  2   END SETUP;
16:03:28 USER
16:03:28 USER   00044  1   SIGNON: PROCEDURE (PORTPTR,MESSAGEPTR,SIZE);
16:03:28 USER
16:03:28 USER   00045  2   /*OUTPUTS SIGNON MESSAGE OF LENGTH  SIZE  TO THE
16:03:28 USER
16:03:28 USER   00046  2    APPROPRIATE PORT*/
16:03:28 USER
16:03:28 USER   00047  2   DCL PORTPTR ADDRESS;
16:03:31 USER
16:03:31 USER   00048  2   DCL MESSAGEPTR ADDRESS;
16:03:31 USER
16:03:31 USER   00049  2   DCL  (MESSAGE BASED MESSAGEPTR) (1) BYTE;
16:03:31 USER
16:03:31 USER   00050  2   DCL (I,SIZE) BYTE;
16:03:31 USER
16:03:31 USER   00051  2    I=0; /*INITIALIZE LENGTH COUNTER*/
16:03:31 USER
16:03:31 USER   00052  2    DO WHILE I<SIZE;
16:03:31 USER
16:03:36 USER   00053  2     IF READY THEN
16:03:36 USER
16:03:36 USER   00054  3     DO;
16:03:36 USER
16:03:36 USER   00055  3      PORT=MESSAGE(I);
16:03:36 USER
16:03:36 USER   00056  4      I=I+1;
16:03:36 USER
16:03:37 USER   00057  4     END;
16:03:37 USER
16:03:37 USER   00058  3    END;
16:03:37 USER
16:03:37 USER   00059  2   END SIGNON;
16:03:37 USER
16:03:38 USER   00060  1   FMUYK:  PROCEDURE INTERRUPT 1;
16:03:38 USER
16:03:38 USER   00061  2   /*FORMS PORT ADDRESS FROM IOINT LATCH THEN TRANSFERS
DATA
16:03:38 USER
16:03:42 USER   00062  2    FROM PORT AND PORT ADDRESS TO AN-UYK20 AS 16-BIT WOR
D*/
16:03:42 USER
16:03:42 USER   00063  2    TEMP(0)=UYK20(0);
16:03:42 USER
16:03:42 USER   00064  2    TEMP(1)=UYK20(1);
16:03:42 USER
16:03:42 USER   00065  2    PORTPTR=MEMMAP + TEMP(1);
16:03:42 USER
16:03:42 USER   00066  2    PORT=TEMP(0);
16:03:42 USER
16:03:42 USER   00067  2   END FMUYK;
16:03:43 USER
16:03:43 USER   00068  1   TOUYK: PROCEDURE INTERRUPT 2;
16:03:43 USER
16:03:43 USER   00069  2    TEMP(1)=IOINT;
16:03:43 USER
16:03:43 USER   00070  2    PORTPTR=MEMMAP + TEMP(1);
16:03:43 USER
16:03:44 USER   00071  2    TEMP(0)=PORT;
16:03:44 USER
16:03:44 USER   00072  2    UYK20(0)=TEMP(0);
16:03:45 USER
```

```
16:03:45 USER    00073  2     UYK20(1)=TEMP(1);
16:03:45 USER
16:03:45 USER    00074  2    END TOUYK;
16:03:46 USER
16:03:46 USER    00075  1    START:
16:03:46 USER
16:03:46 USER    00076  1     TEMP(1)=INPUT(0); /*CLEAR MASTER INTERRUPT CONTROL*/
16:03:46 USER
16:03:47 USER    00077  1     IOINTADR=8F00H;
16:03:47 USER
16:03:47 USER    00078  1     UYKAD=8008H;
16:03:47 USER
16:03:47 USER    00079  1     CALL SETUP(URT2,MODE1,ONLINE);
16:03:48 USER
16:03:48 USER    00080  1     CALL SIGNON(URT2,.MESSAGE,9);
16:03:48 USER
16:03:48 USER    00081  1     CALL SETUP(URT3,MODE3,ERESET);
16:03:49 USER
16:03:49 USER    00082  1     CALL SETUP(URT4,MODE3,ERESET);
16:03:50 USER
16:03:51 USER    00083  1     WAITSFORSINTERRUPT;
16:03:51 USER
16:03:51 USER    00084  1
16:03:51 USER
16:03:51 USER    00085  1    EOF
16:03:52 USER
16:03:52 USER    NO PROGRAM ERRORS
16:03:52 USER
16:03:54 USER    STOP
16:03:55 USER
16:03:55 USER    END OF EXECUTION
16:03:55 USER    CPU TIME: 23.12 ELAPSED TIME: 1:22.50
16:03:55 MONTR   EXIT
16:03:55 MONTR
16:03:55 MONTR   .R PLM82
16:03:55 USER
16:04:00 USER    8080 PLM2 VERS 3.0
16:04:00 USER
16:04:00 USER
16:04:00 USER    $V=9 $C=1 $F=1 $H=64 $$
16:04:01 USER    $A=0
16:04:01 USER    $B=7
16:04:01 USER    $C=0
16:04:01 USER    $D=120
16:04:01 USER    $E=0
16:04:01 USER    $F=1
16:04:01 USER    $G=1
16:04:01 USER    $H=64
16:04:01 USER    $I=1
16:04:01 USER    $J=6
16:04:01 USER    $L=1
16:04:01 USER    $M=1
16:04:01 USER    $N=0
16:04:01 USER    $O=1
16:04:01 USER    $P=0
16:04:01 USER    $Q=1
16:04:01 USER    $R=73
16:04:01 USER    $S=0
16:04:01 USER    $T=1
16:04:01 USER    $U=7
16:04:01 USER    $V=9
16:04:01 USER    $W=72
16:04:01 USER    $Y=1
16:04:01 USER    $Z=2
```

```
16:04:01 USER    S%=0
16:04:01 USER
16:04:01 USER
16:04:02 USER        1=0043H    34=0046H    35=004FH    37=0055H    40=0060H    41=006
5H
16:04:03 USER       42=006CH    43=0074H    44=0075H    47=007DH    52=0080H    53=008
9H
16:04:03 USER       54=0092H    55=0095H    56=00A0H    57=00A9H    58=00A8H    59=00A
BH
16:04:03 USER       60=00ACH    63=00B0H    64=00B4H    65=00BFH    66=00CEH    67=00D
0H
16:04:03 USER       68=00DEH    69=00E2H    70=00EBH    71=00F9H    72=00FCH    73=010
0H
16:04:04 USER       74=010CH    75=0114H    76=0119H    77=011BH    78=011CH    79=012
4H
16:04:04 USER       80=0139H    81=0149H    82=0157H    83=0166H    84=016AH
16:04:05 USER    STACK SIZE = 26 BYTES
16:04:06 USER    MEMORY.....................0A00H
16:04:06 USER    UYKAD......................09ECH
16:04:08 USER    CMDAD......................09EEH
16:04:08 USER    IOINTADR...................09F0H
16:04:08 USER    PORTPTR....................09F2H
16:04:08 USER    TEMP.......................09F4H
16:04:08 USER    MESSAGE....................0046H
16:04:08 USER    SETUP......................004FH
16:04:08 USER    PORTPTR....................09F6H
16:04:08 USER    MDCMD......................09F8H
16:04:08 USER    CDCMD......................09F9H
16:04:08 USER    SIGNON.....................0075H
16:04:08 USER    PORTPTR....................09FAH
16:04:08 USER    MESSAGEPTR.................09FCH
16:04:08 USER    SIZE.......................09FEH
16:04:09 USER    I..........................09FFH
16:04:09 USER    FMUYK......................00ACH
16:04:09 USER    TOUYK......................00DEH
16:04:09 USER    START......................0114H
16:04:09 USER    WAIT.......................0167H
16:04:09 USER    0000H JMP    40H    00H    NOP    NOP    NOP    NOP    NOP    JM
P
16:04:09 USER    0009H ACH    00H    NOP    NOP    NOP    NOP    NOP    JMP    DE
H
16:04:09 USER    0012H 00H
16:04:09 USER    0040H LXI SP ECH    09H    JMP    14H    01H
16:04:12 USER    0046H 24H 50H 4FH 52H 54H 32H 24H 0DH 0AH
16:04:12 USER    004FH LXI H  F8H    09H    MOV MC INR L  MOV ME MOV LI F6H    MO
V AM
16:04:12 USER    0058H INR L  MOV BM ORA I  40H    MOV CA MOV AB ORA I  00H    MO
V LI
16:04:12 USER    0061H EEH    MOV MC INX H  MOV MA MOV LI F8H    MOV CM LHLD   EE
H
16:04:12 USER    006AH 09H    MOV MC LXI H  F9H    09H    MOV CM LHLD   EEH    09
H
16:04:12 USER    0073H MOV MC RET    LXI H  FCH    09H    MOV MC INX H  MOV MB IN
R L
16:04:12 USER    007CH MOV ME INR L  MOV MI 00H    LXI H  FFH    09H    MOV AM DC
R L
16:04:14 USER    0085H SUB M  JNC    ABH    00H    LHLD   EEH    09H    MOV AM AN
A I
16:04:14 USER    008EH 01H    MOV CA XRA A  SUB C  JNC    80H    00H    LXI H  FF
H
16:04:14 USER    0097H 09H    MOV CM MOV BI 00H    LHLD   FCH    09H    DAD B  MO
V AM
16:04:14 USER    00A0H LHLD   F2H    09H    MOV MA LXI H  FFH    09H    INR M  JM
P
```

```
16:04:14 USER   00A9H 3CH    00H    RET    PUSH H PUSH D PUSH B PUSH A LHLD   EC
H
16:04:16 USER   00B2H 09H    MOV AM LXI H  F4H    09H    MOV MA INX H  PUSH H LH
LD
16:04:16 USER   00BBH ECH    09H    INX H  MOV AM POP H  MOV MA LXI H  F4H    09
H
16:04:16 USER   00C4H INX H  MOV AM MOV CA MOV BI 00H    LXI H  00H    80H    DA
D B
16:04:16 USER   00CDH SHLD   F2H    09H    LXI H  F4H    09H    MOV CM LHLD   F2
H
16:04:16 USER   00D6H 09H    MOV MC POP A  POP B  POP D  POP H  EI     RET    PU
SH H
16:04:18 USER   00DFH PUSH D PUSH B PUSH A LXI H  F4H    09H    INX H  XCHG   LH
LD
16:04:18 USER   00E8H F0H    09H    MOV AM STAX D LXI H  F4H    09H    INX H  MO
V A
16:04:18 USER   00F1H MOV CA MOV BI 00H    LXI H  00H    80H    DAD B  SHLD   F2
H
16:04:18 USER   00FAH 09H    MOV AM LXI H  F4H    09H    MOV MA MOV CM LHLD   EC
H
16:04:18 USER   0103H 09H    MOV MC INX H  PUSH H LXI H  F4H    09H    INX H  MO
V AM
16:04:18 USER   010CH POP H  MOV MA POP A  POP B  POP D  POP H  EI     RET    LX
I H
16:04:18 USER   0115H F4H    09H    INX H  XCHG   IN     00H    STAX D LXI H  F0
H
16:04:18 USER   011EH 09H    MOV MI 00H    INX H  MOV MI 8FH    MOV LI ECH    MO
V MI
16:04:18 USER   0127H 08H    INX H  MOV MI 80H    MOV LI F6H    MOV MI 12H    IN
X H
16:04:20 USER   0130H MOV MI 80H    MOV CI 4EH    MOV EI 37H    CALL   4FH    00
H
16:04:20 USER   0139H LXI H  FAH    09H    MOV MI 12H    INX H  MOV MI 80H    LX
I B
16:04:20 USER   0142H 46H    00H    MOV EI 09H    CALL   75H    00H    MOV LI F6
H
16:04:20 USER   014BH MOV MI 13H    INX H  MOV MI 80H    MOV CI 41H    MOV EI 10
H
16:04:20 USER   0154H CALL   4FH    00H    LXI H  F6H    09H    MOV MI 14H    IN
X H
16:04:20 USER   015DH MOV MI 80H    MOV CI 41H    MOV EI 10H    CALL   4FH    00
H
16:04:20 USER   0166H EI     JMP    67H    01H    EI     HLT
16:04:23 USER   NO PROGRAM ERRORS
16:04:23 USER
16:04:23 USER   STOP
16:04:23 USER
16:04:23 USER   END OF EXECUTION
16:04:23 USER   CPU TIME: 10.08 ELAPSED TIME: 28.00
16:04:28 MONTR  EXIT
16:04:28 MONTR
16:04:28 MONTR  .

                !
16:04:23 MONTR  .KJOB DSKC0:TXTOUT.LOC=/W/B/Z:4/VR:10/VS:3304/VL:200/VP:10/VD:P
16:04:29 K-QUE  TOTAL OF 178 BLOCKS IN 3 FILES IN LPT REQUEST
16:04:30 LGOUT  JOB 24, USER [253,533]  LOGGED OFF TTY65    1604  26-JAN-77
16:04:30 LGOUT  SAVED ALL FILES (635 BLOCKS)
16:04:30 LGOUT  RUNTIME 38.22 SEC
```